SciDAC Visualization and Analytics Center for Enabling Technologies
Semi-Annual Progress Report
April 2009 through November 2009

E. Wes Bethel[*] and Chris Johnson[†]
Principal Investigators

Charles Hansen, Valerio Pascucci, Claudio Silva, Allen Sanderson [‡]

Sean Ahern, George Ostrouchov, Dave Pugmire, Jeremy Meredith[§]

Eric Brugger, Peer-Timo Bremer, Brad Whitlock[¶]

Ken Joy, Eduard Deines, Christoph Garth, Bernd Hamann[‖]

Hank Childs, Janet Jacobsen, Prabhat, Oliver Rübel, Daniela Ushizima, Gunther Weber[**]

December 2009

[*] Lawrence Berkeley National Laboratory
[†] Scientific Computing Institute, University of Utah
[‡] Scientific Computing Institute, University of Utah
[§] Oak Ridge National Laboratory
[¶] Lawrence Livermore National Laboratory
[‖] University of California – Davis
[**] Lawrence Berkeley National Laboratory

Contents

1 Executive Summary

The SciDAC Visualization and Analytics Center for Enabling Technologies (VACET) focuses on leveraging scientific visualization and analytics software technology as an enabling technology for increasing scientific productivity and insight. Our mission is to foster scientific insight through creating and deploying effective data understanding technology that is truly responsive to the needs of our stakeholders in the scientific research community who are "awash in data." It is widely accepted that one of the bottlenecks in contemporary science is the need to gain insight from vast collections of complex data.

The vision for our Center is to respond directly to this challenge by adapting, extending, creating when necessary and deploying visualization and data understanding technologies for our science stakeholders. Organized as a Center for Enabling Technologies, we are well positioned to be responsive to the needs of a diverse set of scientific stakeholders in a coordinated fashion using a range of visualization, mathematics, statistics, computer and computational science and data management technologies.

We are pleased to report accomplishments during the period of April 2009 through November 2009, both in terms of impact for scientific stakeholders and in terms of providing leadership in the visualization and analysis community.

The VACET team has conducted a performance study of the VisIt visualization application to better understand its performance at very high levels of concurrency and on very large data sets. These experiments consist of a weak scaling study of a production-quality visualization code (VisIt) on six different platforms on the largest-ever data sets published in open literature. This result is significant because it shows that VACET's production-quality visualization software is capable today of handling tomorrow's scientific simulation data sets on DOE's largest computational platforms (Section 4.1).

The VisIt visualization application is now an official DOE/ASCR Joule code after passing the Joule software scalability metric. The Joule program within DOE ASCR is an important metric to ensure that codes seen as important to the DOE ASCR's mission are making effective and efficient use of HPC resources. For 2009, VisIt was chosen as a Joule program candidate. In October of 2009, VisIt passed the Joule metric and is now an official Joule code. VisIt is the first ever non-simulation code to be selected for Joule code certification (Section 4.2). A side benefit of this accomplishment, as well as the VisIt "hero runs" work, is a close and productive relationship between VACET's SciDAC visualization work and investments elsewhere in DOE, namely NNSA.

VACET continues to perform meaningful work that has broad impact in the science community:

- Accelerator modeling. VACET researchers devised a new process for performing analysis of beam paths in simulations used to model laser-wakefield accelerators (Section 2.1). This work resulted in numerous technical publications co-authored by science stakeholders.
- Applied mathematics. The SciDAC Partial Differential Equations Center (APDEC) has adopted VisIt as its community-centric AMR visual data exploration and analysis tool. This move requires ongoing support in the form of feature enhancements, performance optimization, bug fixes, consulting and outreach (Section 2.2).
- Astrophysics. VACET's focus on field-leading AMR visualization requires that we provide consulting/training and outreach to AMR code stakeholders, like those in the SciDAC Computational Astrophysics Consortium (Section 2.3).
- Climate. Working with the SciDAC Earth Systems Grid, our team is providing important new visual data analysis and exploration capabilities to the climate research community. Recent

VACET work has resulted in new tools used to produce images and moves that were shown at the December 2009 climate summit in Copenhagen, Denmark (Section 2.4.1).

- Combustion. VACET researchers conduct field-leading research in the area of topological analysis of combustion simulation data produced by AMR codes (Section 2.5.1) and direct numerical simulation (Section 2.5.2).

- Fusion. Our team is providing assistance to six different SciDAC Fusion projects by focusing on key needs common to all project areas. The new capabilities are being developed and deployed in VisIt (Section 2.6).

Part of VACET's technical portfolio includes visualization research. Recent results have resulted in a novel approach for performing integral curve calculations – the fundamental algorithm used for many vector field visualization techniques, including streamlines – in parallel, which resulted in a publication in the SC09 technical program and that has been deployed in production form in VisIt and publicly released for widespread use (Section 3.1). Other recent work includes application of topological analysis technology to two different combustion science problems to better understand the relationship between turbulence and combustion characteristics (Section 2.5.1) and to help gain a deeper understanding about the machanisms of reignition (Section 2.5.2).

VACET continues to successfully transition field-leading research into production quality software for use by the scientific research community. To address the need for faster and higher-quality volume rendering technology, the team has performed an initial port of the Tuvok volume rendering library into VisIt with an expected public release in CY2010 (Section 4.4). Related work focuses on porting the IceT compositing infrastructure into VisIt so as to efficiently perform sort-last compositing, an algorithmic step that enables effective use of large, distributed-memory GPU clusters for use in production-quality scientific visual data analysis and exploration (Section 4.5). Other technology transfer activities include: deploying topological analysis capabilities – Reeb Graph computation – into the open source VTK library (Section 4.8); adding a VisTrails plugin for VisIt so that science users can benefit from the combination of workflow/provenance technology with production-quality, parallel capable visual data exploration and analysis infrastructure (Section 4.7); providing the climate science community with software tools for comparative visual data analysis and exploration as well as provenance and workflow management (Section 2.4.2).

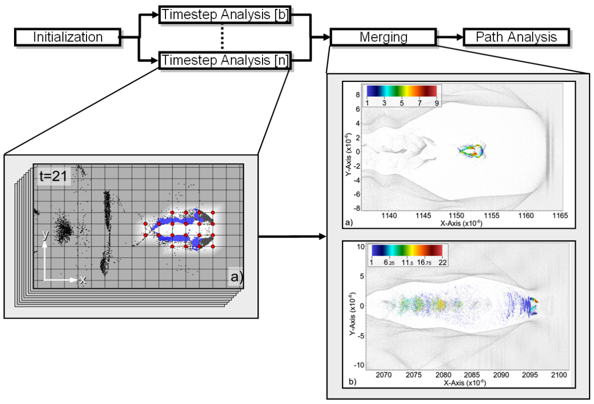
During this period, VACET has had a prolific number of technical publications: 19 peer-reviewed journal articles, 13 peer-reviewed conference proceedings, two invited articles, one poster, two books, eight book chapters, and four technical reports. We have provided content for six press releases (Section 5).

We have continued our aggressive outreach campaign (Section 6): 18 invited presentations and eight different tutorials. The tutorials we presented at VisWeek in October 2009 (IEEE Visualization 2009) received glowing press in the visualization community, which helps to broaden the audience for our visualization effort and helps us to maintain our leadership position in the field.

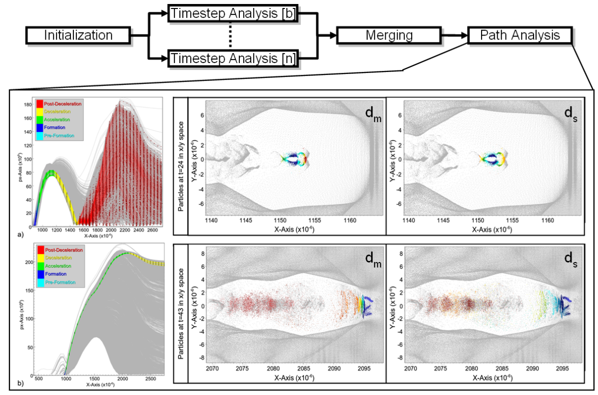
2 Specific Stakeholder Projects

2.1 Accelerator

The main objective for this project is to develop methods for the analysis of laser wakefield particle accelerator simulation data. Currently we focus on development of methods for detection and analysis of particles beams. This involves development of advanced methods for visual exploration



(a) We first segment particle bunches at individual time steps using an efficient grid-based analysis, then merge the results from the individual time steps (i) to define the number of detected bunches and (ii) to consolidate the description of the different bunches.



(b) In the final analysis step, we analyze the complete temporal paths of all particles identified in the previous analysis steps.

Figure 1: Illustration of the general structure of the new beam path analysis pipeline. In this example we detect two bunches. We compute for each bunch the different temporal phases of the bunch — when was the bunch formed, accelerated, then outrun the plasma wave, then finally decelerated — as well as two different distance fields: (i) the distance in physical space d_s and (ii) distance in momentum space d_m between of each particle to the bunch.

of the particle data as well as methods to automate the most time-consuming steps of the analysis process.

High-performance Visual Exploration of Particle Data.

Laser wakefield accelerator simulation data sets contain several hundred million particles per time step, while only a small fraction of these particles are accelerated to high-energy levels and are part of a particle beam of interest. To enable fast visual data exploration, efficient particle tracing, and interactive beam selection, we focused in the previous reporting period on integrating the data management system FastBit with VisIt. In this reporting period we have continued this effort and deployed the developed functionality in VisIt (see Section 4.3) and investigated the applicability of this system to fusion data (Section 2.6).

Beam Path Analysis.

The main objective of this project is to develop methods for automatic detection of particle beams in laser wakefield particle accelerator simulation data and classification of their temporal behavior. Detection of particle bunches of interest is performed manually at present. Our capability helps to largely automate this process and increase the accuracy of the selection process. Currently, particle bunches are classified based on a single reference time step. The project aims to use the complete temporal history of the particles to enable more accurate beam classification. The large sizes (TBs) of current data sets presents challenges for the analysis of the time series. One main objective of this project is therefore the development of efficient data analysis and data management methods (such as FastBit).

Analysis of the temporal evolution of particle bunches is essential for the understanding of how particle beams are formed and accelerated in a laser wakefield accelerator. In the long run, this capability will certainly help accelerator scientist to accelerate data understanding. We are also planning to integrate this method with other Visualization/Analysis tools (VisIt) as part of a broad set of high-performance tools for scientific knowledge discovery.

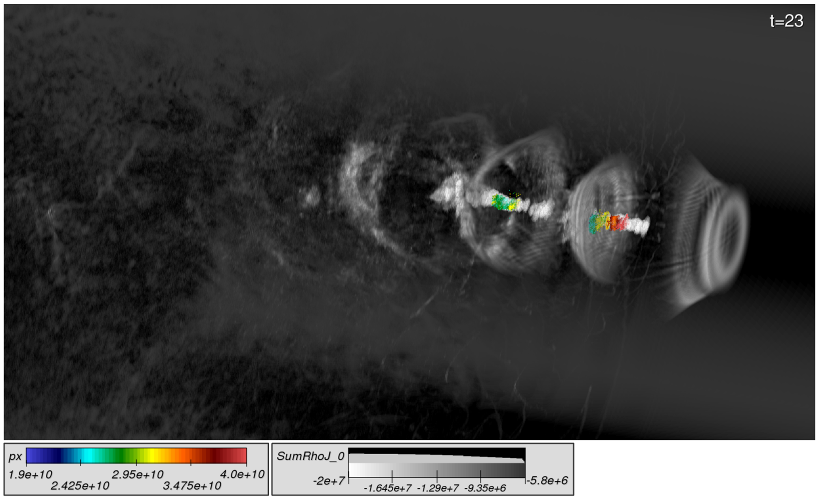


Figure 2: This image shows two particle bunches (color) detected automatically by the beam path analysis in a large 3D data set. A volume rendering of the plasma density (gray) shows the structure of the plasma wave, illustrating the location of the two bunches within the wave.

Accomplishments this period:

We developed a new algorithm for automatic detection and classification of particle beams. Figure 1 provides an overview of the proposed analysis pipeline. In the Initialization phase, we load meta-data and compute a minimum time point to identify the time steps relevant for the subsequent analysis. Afterwards, we perform segmentation of individual particle bunches in the Timestep Analysis step of the pipeline. We then merge the results from the individual time steps to define the number of different particle bunches and define a consolidated description of each bunch. In the final Path Analysis, we first trace all detected candidate particles over the complete time series. Based on the information from the Merging and the particle tracing, we compute a reference path for each bunch and define the different temporal phases of a bunch (e.g., formation, acceleration, deceleration). Finally, we compute for each candidate particle the distance of its path to the reference path of the respective bunch.

We developed visualizations of analysis results and intermediate analysis steps in order to validate analysis results and to define thresholds for the computed distances fields to accurately define the bunches of interest. We developed custom routines to dump VTK files of results from all steps of the analysis pipeline. Visualization of analysis results is then done using VisIt. Visualizations used for validation of the analysis include: (1) Scatter/Pseudocolor plots to show the structure of analysis results in 2D and 3D spaces using point based visualizations; (2) Particle trace plots to investigate the temporal behavior of particle bunches; (3) Surface plots to validate the segmentation procedure (Step 2 of the analysis pipeline); and (4) 1D/2D Histograms to investigate the distribution of the computed path distance fields.

To achieve high performance, we leverage FastBit for computing 3D, conditional histograms and tracing of particles over time. Our analysis pipeline makes heavy use of 3D histograms in particular for detection and segmentation of particle bunches at individual time steps. In close collaboration with the SciDAC SDM Center, we have worked on the development of efficient methods for computation of 3D histograms. Over the course of the project, we worked closely with the SDM Center to provide them with requirements for FastBit histogram computation and studied the performance of different methods for histogram computation. The SciDAC SDM Center implemented the required extensions to FastBit to support our needs, which we deployed and tested

in our code. There are two main new features in FastBit that we requested and have subsequently deployed in our code. First is the computation of 3D, conditional histograms. Second are FastBit extensions that compute counts for a 3D histogram and also a set of bitvectors to enable fast access of data portions associated with a set of histogram bins. This new functionality is required to enable efficient implementation of Step 2 of the analysis pipeline. Future integration of this type of analysis with VisIt will rely heavily on the successful integration of FastBit into VisIt (Section 4.3), which is one of our infrastructure projects benefitting this accelerator project as well as projects for fusion customers.

We conducted an extensive performance evaluation of the beam path analysis pipeline and tested it with several different data sets, including a very large ($\approx$ 610GB) 3D particle data set (see Figure 2). Even in the case of the 3D data set, the analysis took only $\approx$ 185 seconds in serial.Up until recently, 3D tagged data sets (i.e., 3D data with unique particle Ids) have not been available and we are planning to conduct more detailed studies in 3D in the future.

We validated the analysis pipeline by executing it on several different data sets and evaluated results with our collaborators who are part of the SciDAC Community Petascale Project for Accelerator Science and Simulation (COMPASS).

This work resulted in a publication in the Institute of Physics Journal of Computational Science and Discovery [64].

Future work for this project includes:

- Parallelization of the proposed analysis pipeline.
- Improvement of visual presentation of analysis results.
- Development of methods for comparative analysis of particle beams.
- Development of new analysis algorithms.
- Integration of the analysis into VisIt for more widespread deployment.
- Investigation and analysis of simulations using colliding laser pulses.

2.2 AMR Visualization and the SciDAC Applied Partial Differential Equations Center

Our primary AMR stakeholders—the SciDAC Applied Partial Differential Equations Center (APDEC) and the LBNL Center for Computational Science and Engineering (CSEE)—have need for production-quality AMR visual data analysis infrastructure. Both teams have an in-house tool they have been using and funding over the years. Both teams have expressed the urgent need to get out of the business of developing and maintaining such software infrastructure. Additionally, both teams have expressed the need for new capabilities that lie outside the scope of what is possible with their existing technology.

One notable strength of VACET's approach to AMR visualization is that by teaming with APDEC and CCSE, we enjoy the benefit of reaching all of their science collaborators. Both APDEC and CCSE are best-in-field in AMR-based computational science. A good example of this "inherited customer base" is through the progress we make with their customers. One specific example in this report is our work with the SciDAC Computational Astrophysics Consortium: they rely on CCSE to provide AMR codes for modeling supernova explosion. See Section 2.3 for more information.

The activities to support these teams can be described with five high level thrusts:

- Invest in critical, missing visualization algorithms.

- Planned interface changes to ease the transition of their user community.
- Planned functionality improvements specific to APDEC.
- Maintenance of software infrastructure, bug fixes, and other short term, unforeseen changes.
- Performance improvements specific to AMR data sets.

A more detailed list of specific needs is documented on the VACET twiki[1]. That list contains around forty different features requested by customers from APDEC and the Astrophysics SAP project. Because the tasks are so numerous and since many will take a significant amount of time to accomplish, our approach is to prioritize these features to optimize the intersection between (1) the stakeholder's assessment of the feature's importance and (2) the degree of difficulty in implementing the feature. With this prioritization, we order our work to achieve impact quickly while continuing work on more difficult and challenging research and engineering problems. The tasks of each of the four categories are itemized below along with labor efforts, stakeholder priority, and status (in some cases where appropriate).

Accomplishments this period include:

- Adding support for particle data in Chombo files (existing format, there are plans to define a new format). Note: This item is an example of a need that was addressed as it came up and was not listed in a previous task list.
- Due to a security issues, APDEC switched from the OpenSUSE Linux distribution to Ubuntu. We set up a new build environment for VisIt on their Ubuntu machines and provided them with a customized VisIt build for their machines. Note: This item not on the original list of stakeholder requirements as it was outside the field of view during early planning.
- Provided continuously updated customized VisIt builds on APDEC machines including build of the current developer version for evaluation of new features (e.g., embedded boundary support) by APDEC group members.
- Fixed bug in Chombo reader that caused incorrect cycle information to be displayed. Note: This is something that was fixed as it came up and not part of the original list of requirements.
- Improved mapped grid support by adding the ability to map 2D Chombo data to 3D.
- Improved VisIt's Laplace operator to only require one layer of ghost cells.
- Honor material selections in Vector plot and only display vector glyphs corresponding to active materials in mixed cells.
- Fixed a network bug where VisIt on a laptop freezes when the network connection is changed.
- Fixed VisIt crash on directory listing when the directory contains "numbered files" where numbers only vary in the number of digits (e.g., if there are two files plot005.hdf5 and plot00005.hdf5 in the directory).

[1] A complete list of APDEC AMR visualization needs is documented here: `http://www.sci.utah.edu/vacetwiki/index.php/Collab:APDEC_VisIt`.

- Improved support for Embedded Boundary extraction (see Section 3.2). We have new schemes that together should resolve most of APDEC problem. In addition to the tests described in Section 3.2, we need to verify that the performance enhancements are sufficient and that there are no new performance issues introduced by the new algorithms.

- Fixed "visitrc bug" where custom macro buttons for APDEC sometimes do not get properly added. However, a sleep command to avoid a race condition still seems necessary. Need some of Brad Whitlock's time to resolve. (However, Brad uses a Mac and this seems a Linux only problem.)

- Investigated fix to a "runaway process" problem where VisIt processes did not get terminated properly. (However, it seems that sometimes there are still runaway engines on daVinci. Possibly check with NERSC how common the problem is.

- Optimizations for large numbers of patches in AMR data sets. Due to conflicts with the NVIDIA driver we needed to disable the `tcmalloc` enhancements. Since glibc memory management has improved considerably and all APDEC machines uses a recent Ubuntu release, using `tcmalloc` may no longer be necessary.

- Streamlines are ongoing work jointly with APDEC researchers. We have a first prototypical implementation and are evaluating it. This is a bit more open ended, since the solution is not available in a regular VisIt build, yet. See Section 3.1 for more details.

- Day-to-day consulting and assistance for APDEC and CCSE stakeholders.

- Outreach: mini-tutorial (90 mins) for four UC Davis students and Greg Miller (Chombo customer).

Work plans for the future include:
High Priority Items

- Finish evaluating partial solutions listed above (except streamlines which are probably a bit more open-ended.)

- Deploy some form of AMR Streamlines to APDEC for testing.

- Implement native double precision support in VisIt pipelines. This item is a target for VisIt 2.0 release in Feb/Mar 2010.

- Fix bug that occurs in developer build after transition to Qt 4 where macro buttons are invisible if a window is "posted" at startup.

Medium Priority Items

- Implement crack-free isosurfaces. Priority for this task has been increased due to increasing stakeholder requests.

- Implement scalable selection of patches in an AMR data set. The current approach of presenting a list of patches in the GUI and allowing users to turn them on or off via a check box is very tedious for simulations containing a large number of patches.

- Fix intermittently missing patches data sets that contain embedded boundaries (i.e., multiple materials).

- Fix bug where the “Subset plot” (i.e., the plot showing the boundaries of individual patches in a data set) fails with a “BadIndexException” when used on an embedded boundary Chombo data set and not all materials are turned on.
- Fix issues when opening compressed files: (i) Support bzip2 file format; (ii) decompress all time steps of an animation and not only the first; (iii) support multiple instances of VisIt opening the same file without conflicts.

Lower Priority Items

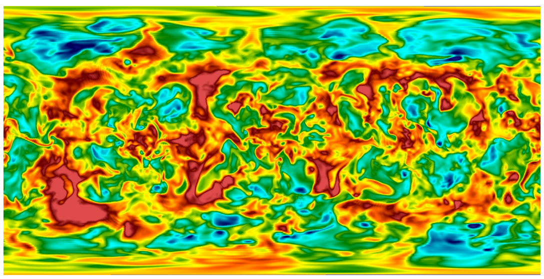
- Fix missing boundaries bug where a “Mesh” plot erroneously does not display partially shared grid faces..
- Inconsistent material selection bug. This may be a non-issue since we now mostly use Chombo ghost data, which should avoid the bug.
- Fix labels for scatter plot.
- Fix bug when reopening compressed files.
- Make it possible to use a slider to animate slices through a data set.

2.3 SciDAC Computational Astrophysics Consortium

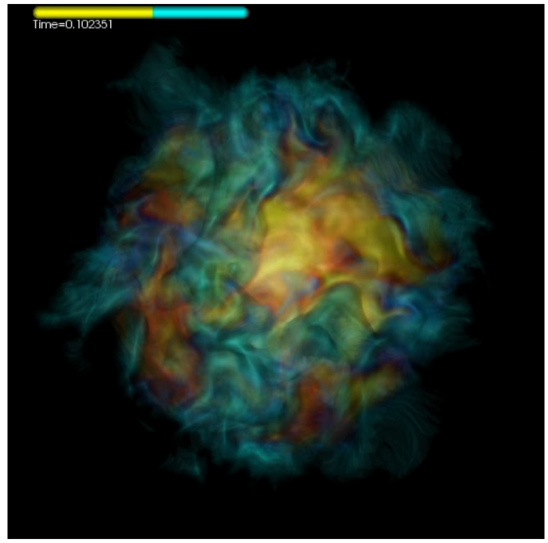
This project aims to provide direct support to Adam Burrows (Princeton), who is part of the SciDAC Computational Astrophysics Consortium (CAC). Burrows is migrating away from the older VULCAN code to the newer CASTRO code, which is a product of the LBNL Center for Computational Sciences and Engineering (CCSE). CASTRO is an AMR code that outputs AMR data. VACET’s work with AMR stakeholders (Section 2.2) includes direct, one-on-one support. Our work with Burrows in this section is a great example of the type of one-on-one support that is required to help scientists transition from one code and visualization tool to another. While our work is one-on-one with Burrows, the enhancements and bug fixes we make to VisIt will benefit AMR and astrophysics users in general.

Accomplishments this period include:

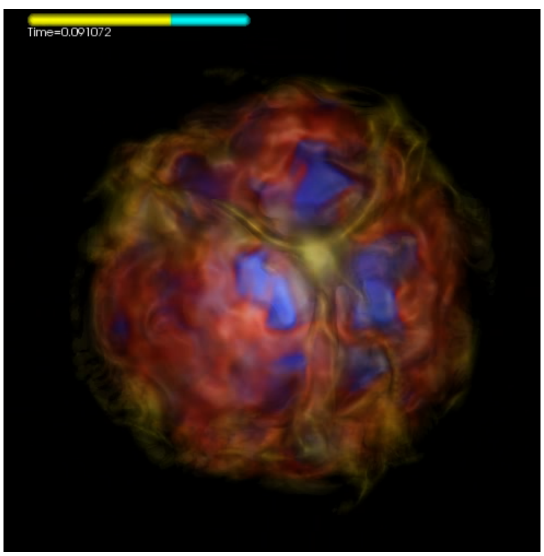
- Provide tutoring, scripts, movies, etc to enable users to use VisIt. This task includes multiple face-to-face visits at Princeton along with a tutorial at Princeton, as well as VACET-produced movies and images, with VisIt scripts being handed off to visualization support personnel at Princeton. See Figure 3 for examples.
- Fix bugs encountered by users. (1) The “hot dog problem:” resampling AMR data has strange artifacts; (2) several streamline problems were encountered, it is believed that this problem has been repaired fixed an issue with streamlines for time varying data, but that has not yet been confirmed.
- We are developing algorithms to quantify the duration neutrinos reside in the star center.



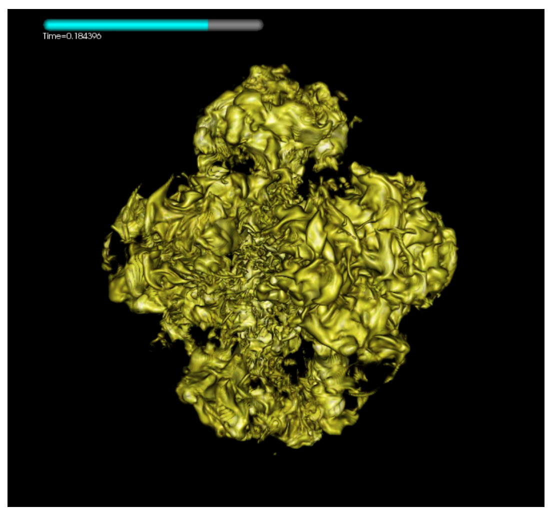
(a) A radial slice of a CASTRO calculation shown on the "globe display" at the LBNL booth at SC09.



(b) Image, created by VACET, from a movie of CASTRO simulation output for Adam Burrow's team showing the variable entropy.



(c) Image, created by VACET, from a movie of CASTRO simulation output for Adam Burrow's team showing the variable Ye.



(d) Image, created by VACET, from a movie of CASTRO simulation output for Adam Burrow's team showing radial velocity.

Figure 3: Images created by VACET for A. Burrows, who is part of the SciDAC Computational Astrophysics Consortium, as part of our ongoing user support mission.

2.4 Climate and the Earth Systems Grid

2.4.1 Advanced Visual Data Exploration and Analysis

The most advanced climate modeling systems seek to enable a new deeper understanding of the dynamics of global carbon cycle, atmospheric chemistry, land and ocean ecological processes and their coupling with climate. This will allow pursuing reliable answers to fundamental questions related to climate variability and global change at time scales ranging from decades to centuries. In this effort VACET will work in close collaboration with the Earth Science Grid and provide new advanced data analysis and visualization tools to the CCSM Consortium and the climate modeling community in general. One target will be the deployment of a first set of tools by FY08, in time to facilitate the analysis of data for the 5th assessment report of the Intergovernmental Panel on Climate Change (IPCC). VACET is committed to supporting the needs of the Community Climate System Model (CCSM) Consortium in collaboration with the Earth System Grid (ESG).

The high level needs of our climate stakeholders are as follows. Deploying advanced visualization capabilities into the CDAT tool and create a clear path for similar integration in other tools. Extending the visualization software to incorporate domain specific requirements, data formats, and vector field visualization. Supporting time-dependent and cross-data set comparison, visualization and analysis. Developing new analytic capabilities for climate data (first deployed into CDAT/VCDAT). Integrating with VisTrails advanced framework for tracking and logging of internal state and provenance information. Developing a visualization and data analysis scenario for understanding of complex coupled phenomena such as the multi-scale dynamics the complete carbon cycle on earth.

This long-term project involves numerous milestones and deliverables that span the five-year VACET project term that have been documented in previous VACET progress reports and management plans. Accomplishments on these goals this period have focused on enhancing the multi-data set and time-dependent capabilities of ViSUS/CDAT. The following list include the activities in progress or completed during this reporting period:

- Added time-dependent loading of data through CDAT allowing arbitrarily sized data sets.
- Added structure to allow linked view exploration by mapped interactions.
- Added time-dependent interface to easily navigate large time sequence.
- Added movie creation interface allowing interactive scene setup and frame rendering.
- Added consistent dates/times through the infrastructure allowing human-centric selection of time steps.
- Added arbitrary time interpolation to increase temporal resolution.
- Participated in SC09 Bandwidth Challenge competition (finalist).
- Prepared images and movies/animations for display at the December 2009 climate summit in Copenhagen [2]. We reported this result to our program manager, who subsequently added a news item about the accomplishment on the `www.scidac.gov` website. Figure 4 shows one frame from this movie.

Future work targets include:

[2] Download the full movie from `http://www.sci.utah.edu/~pascucci/tmp/climate_video/climate_video_test9.qt`.

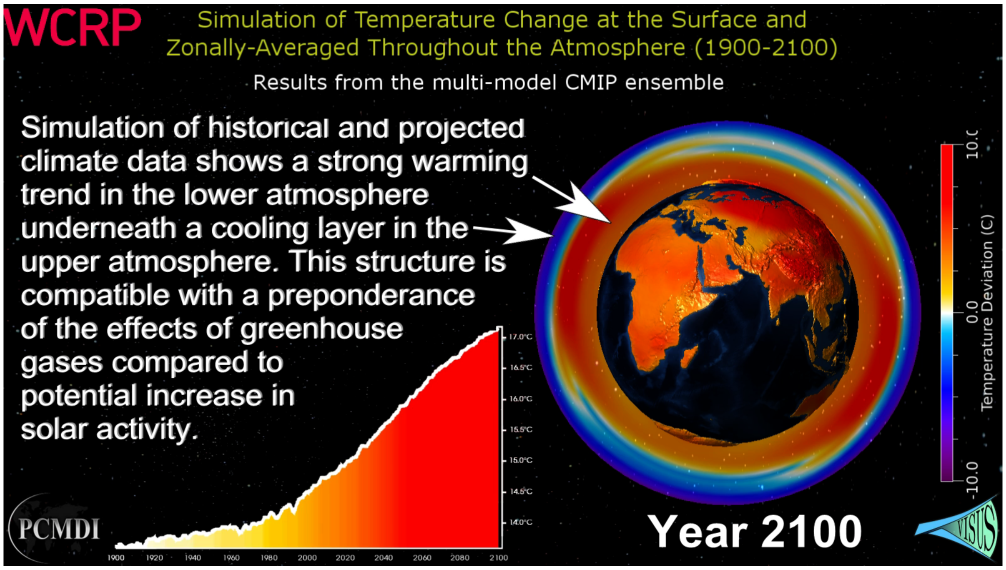


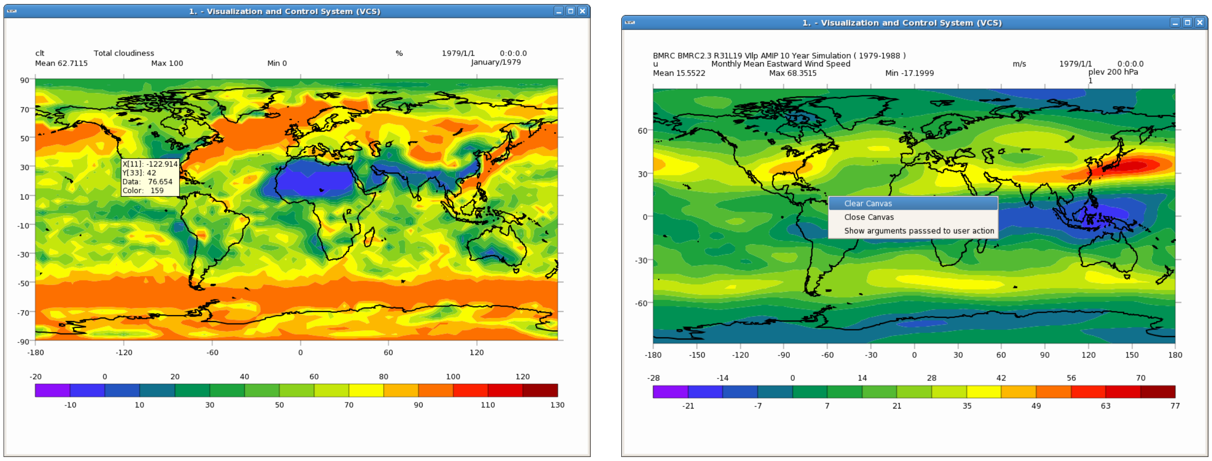
Figure 4: This image is one frame from a movie created by VACET for display by climate researchers at the December 2009 climate summit meeting in Copenhagen.

- Prepare ViSUS for the release of CDAT 6.0.
- Move to Qt based interface which will become the standard.
- Extend functionality as requested by the climate community.
- Re-integrate volume rendering into ViSUS 2.0.
- Develop climate specific user interface.
- Provide example scripts as tutorials for new users.
- Automatic scripts to convert climate data including the appropriate meta-data into idx format.
- Define/develop Qt-based interface for climate analysis.
- Remote visualization client.
- High quality movie output (povray).
- Tutorials.

2.4.2 Provenance and Comparative Visual Analysis

We are working with the ESG group (led by Dean Williams) to provide climate researchers worldwide with access to: data, information, models, analysis, visualization tools, and computational resources required to make sense of enormous climate simulation data sets. Our work will help integrate distributed data and computers, high-bandwidth wide-area networks, and remote computing using climate data analysis tools in a highly collaborative problem-solving environment.

We are adding provenance support and comparative visualization tools to the widely used Climate Data Analysis Tools (CDAT) software infrastructure to help users to understand and



(a) VCS Windows using Qt as the backend and left-click mouse event.

(b) VCS Windows using Qt as the backend and right-click mouse event.

Figure 5: Screen shots showing results of accomplishments porting backend drawing and event handling from X11 to Qt.

interpret their scientific workflows. CDAT uses Python to link together separate software subsystem and packages forming an integrated environment for solving model diagnosis problems. Managing and keeping detailed provenance of CDAT workflows will allow users to examine all the steps that led to a result, identify the experiment's inputs and outputs and consequently help reproduce the results.

Accomplishments during this reporting period:

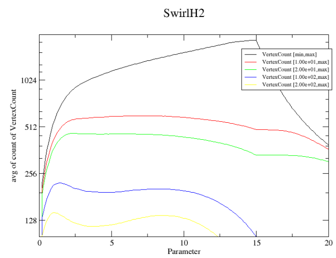
- An analysis of requirements for the translation of python scripts to VisTrails provenance is in progress.

- There was major work on porting drawing backend and event handling from X11 to Qt, and it is close to being finished. The porting provided simplification of parts of the code. Mouse events were also ported (see Figure 2.4.2). Currently the team is trying to solve some major issues with thread management in Mac OS X. In order to complete this milestone, the team has continually collaborated by e-mail and by regular conference calls. Also in July 2009, Charles Doutriaux (from LLNL) visited Utah in order to work close together with Huy Vo. The visit was very productive and led to significant progress.

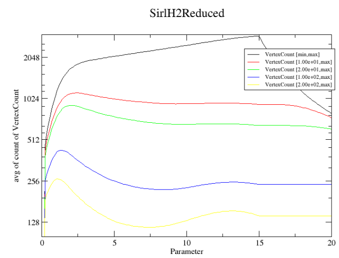
2.5 Combustion Science

2.5.1 Analysis of AMR Combustion Simulation Results

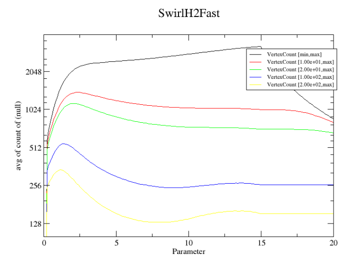
Our team has been working with John Bell (LBNL) and researchers at LBNL's Center for Computational Sciences and Engineering (CCSE) to perform research, development, and application of topological analysis techniques to provide new insights into the combustion process. Bell's group works with large AMR 3D time-varying combustion simulation data sets, and needs analysis tools to aid their understanding. They are interested in formation and evolution of "extinction zones." They may be able to provide us with a Boolean volume of "burning" and "non-burning" regions, and we will perform a detailed analysis of the structural characteristics of the network of low temperature regions. They are also interested in using topological methods to identify those zones with



(a) Number of cells vs fuel consumption threshold for the SwirlH2 data set.



(b) Number of cells vs fuel consumption threshold for the SwirlH2Reduced data set.



(c) Number of cells vs fuel consumption threshold for the SwirlH2Fast data set.

Figure 6: Plots showing results of analysis to compute cell count conditioned on cell size for three different datasets.

topology-based methods. To this end we will use topological tools both for feature characterization and for robust time tracking.

Our work has resulted in a quantitative analysis that has enabled them to see, for the first time, the relative impact of turbulence intensity on cellular burning structures in lean premixed hydrogen flames

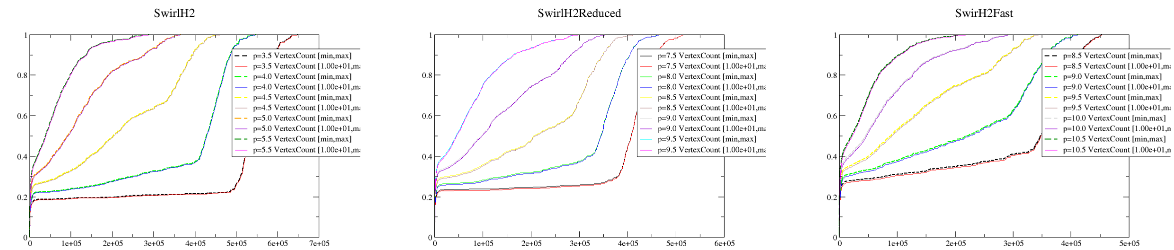
In this reporting period, we are building on previous work to produce a deeper analysis. We have successfully moved to a fully 3D analysis working directly from the native AMR data format. Furthermore, we have implemented a new analysis framework to analyze the initial function (fuel consumption) as well as various derived quantities, e.g. volume of the features, higher order statistical moments of the function, etc. This information is computed on the fly and integrated into a console based statistical analysis framework. Using the new tools we have performed an extensive analysis of three simulations called SwirlH2, SwirlH2Fast, and SwirlH2Reduced. These aim to simulated a device scale low-swirl burner where each run uses different flow profiles. The fuel in this simulations enters the domain in a swirl motion through a circular area of about 2-2.5 cm radius and is surrounded by a uniform coflow of cold air. Furthermore, to help analyze the data we have produced the prototype of an interactive exploration tool which allows to browse a tracking graph while interactively adjusting thresholds and sub-selecting features.

Cell Count Analysis. In previous versions of this analysis we found that for the center region of the SwirlH2 flames the cell count vs. threshold graph no longer showed a plateau for low thresholds. This was of particular concern for the stakeholders as he plateau was used to justify a single "master-threshold" for subsequent analysis. Unfortunately, the analysis of the complete data (including the outside fringe regions of the flame) re-enforced this result. As can be seen in the black curves of the graphs below even a log-plot shows no discernible feature.

However, the plots shown in Figure 6 also show several other curves that indicate the cell count conditioned on cell size. In particular, the count is shown when counting only cells with more than 10, 20, 100, and 200 voxels. Interestingly, these conditioned plots show strong features and in fact their two maxima seem to indicate regions of interest. We are currently engaged with the stake holders to better understand these new results.

Cell size distribution. As before the stakeholders are not only interested in the cell count but also in the distribution of cell sizes. Somewhat surprisingly, as shown below in Figure 7, the conditioned distributions show no noticeable change when compared to the unconstrained plots.

Publications. This work has resulted in four significant publications in which the science stakeholders are co-authors. These publications are listed in the publications section (Section 5) of



(a) NCDF of cell size of the SwirlH2 data set.

(b) NCDF of cell size of the SwirlH2Reduced data set.

(c) NCDF of cell size of the SwirlH2Fast data set.

Figure 7: Plots showing results of analysis to compute normalized cell size distribution function (NCDF) for three different datasets.

this document; the stakeholders are M. Day, J. Bell, V. Beckner, and M. Lijewski.

Work targets for the future include: further exploit the new conditional statistics to better understand the data; rewrite statistics interface to be more general; finish prototype exploration tool and solicit feedback; apply conditional statistics to new data sets; develop tool to quickly and easily browse one-parameter families of statistics; integrate statistics into time-tracking.

2.5.2 Analysis Combustion Direct Numerical Simulation Results

The stakeholder for this work is Jacqueline Chen (SNL-CA) and her team, who work with large 2D and 3D time-varying combustion simulation data sets. They need analysis tools to aid their understanding of the combustion process. They are interested in analyzing the data to detect the dominant re-ignition mechanism of extinction pockets. Our approach is to develop and deliver topological analysis software to aid understanding of the combustion process, in particular, to track and analyze the formation, evolution, and extinction of these features. We focus on the use of robust topological methods for segmentation and tracking of regions of high scalar dissipation rate that provide a good first order approximation of true extinction regions.

In this period, our work has focused on developing new code for computing the Morse-Smale complex for segmentation of OH basins, and new results in identification of ridges.

Identification of High Scalar Dissipation Structures Using MS Complex Decomposition. We focus on data from the JET simulation, a temporally-evolving turbulent CO/H2 jet flame undergoing extinction and reignition at different Reynolds numbers. The simulations were performed with up to 0.5 billion grid points. The configuration is shown on the left of Figure 8 below. Periodic boundary conditions in the mean flow (x) direction results in a situation where the mixing rates increase until approximately midway through the simulation, after which point they begin to decay.

We have implemented a new code base for computing the Morse-Smale (MS) complex required for this segmentation. The new code is designed to be parallelizable, which will allow in-situ computation of statistics. Furthermore, the new code maintains gradient information, allowing the extraction of higher dimensional manifolds. Figure 9 illustrates the identification of the centers of the basins of mixture fraction. We have computed some preliminary statistics characterizing these regions, and will produce further length-scale analysis.

This code base has new interface elements (Figure 10), where feature selector widgets are created on-the-fly as a user changes a feature identification script. This ability to represent variables as free parameters that can be modified in an interaction session has been invaluable in identifying

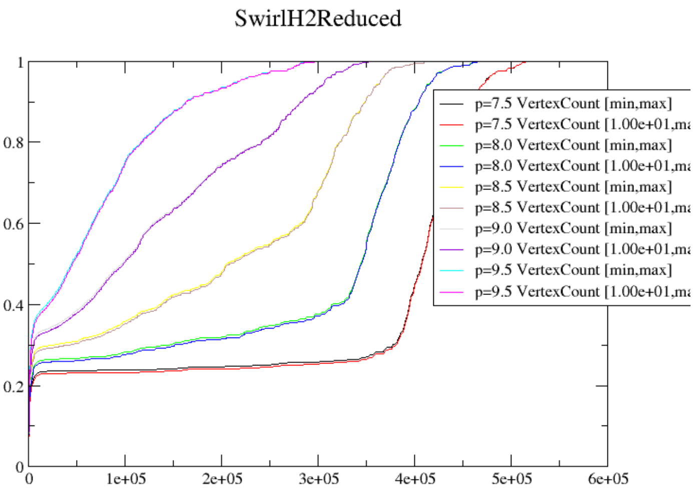


Figure 8: We derive a number of relations among the species of the simulation that allow formulating hypotheses of the dynamics of the extinction and re-ignition process. In particular we compute the Length Scales from statistics of the thickness of the pancake-like scalar dissipation features. Thickness is known to increase over time in JET. However, full characterization of the dissipation feature involves other structures as well, and we compute basins in the mixture fraction field to correlate with the pancake-like feature in scalar dissipation as well as high vortical regions in the velocity field.

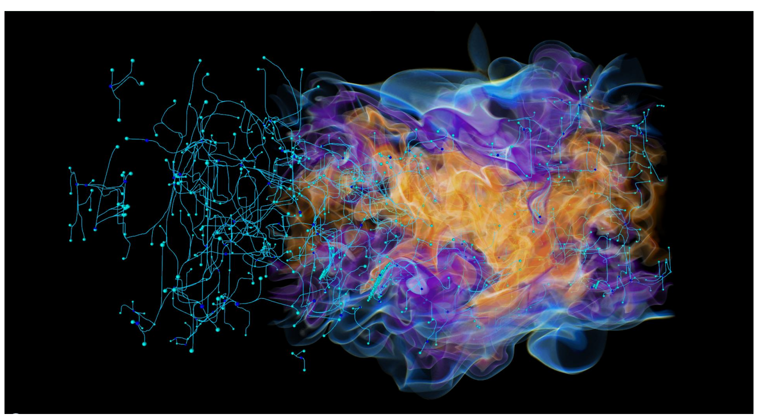


Figure 9: This image shows identification of the centers of the basins of mixture fraction in DNS combustion simulation results.

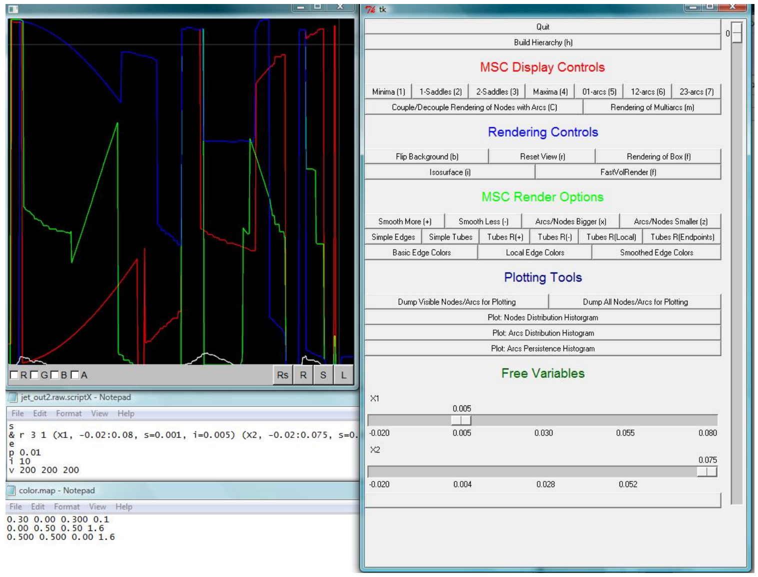


Figure 10: This image shows identification of the centers of the basins of mixture fraction in DNS combustion simulation results.

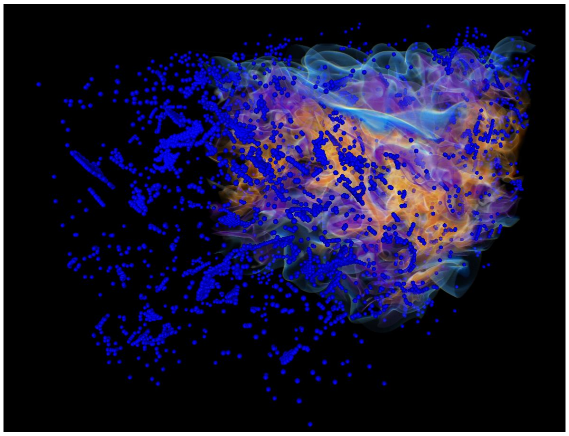
the thresholds for simplification and extraction of the relevant basins.

These thresholds guide topological simplification, removing critical points that only exist due to noise, for example, extracting the lowest minima in basins. Figure 11 illustrates the results of this topological cleaning.

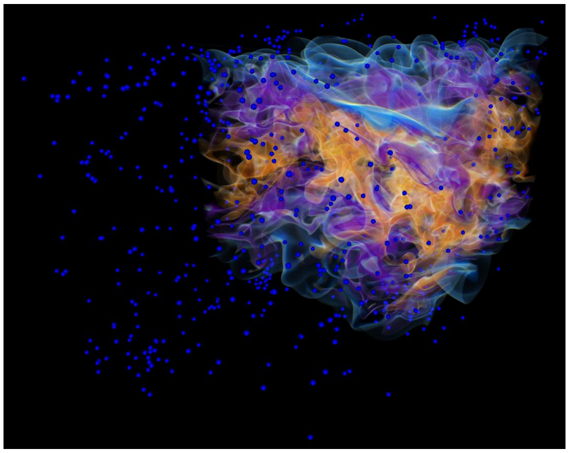
High Scalar Dissipation Structures as Ridge-lines So far we have segmented high scalar dissipation regions as level sets surrounding maxima in the "chi" field. However, for two-dimensional experimental data the traditional method of extracting and analyzing similar structures (using the magnitude of temperature gradient as a stand-in for chi) exist based on computing "ridge-lines." In this context, a point is classified as "ridge" if it is a local maximum in the direction orthogonal to the local gradient (Figure 12). Unfortunately, this criterion is not stable and multiple heuristics are required to achieve an adequate segmentation. Furthermore, initial experiments in 3D using a generalized definition of a ridge have failed to produce useful segmentations. Nevertheless, comparing topological techniques with traditional methods is an important step to validate the new methods. Additionally, replicating ridge-like structures in two-dimensional data using topological techniques may provide important insights into the unsolved problem of computing "ridge-surfaces" in the volumetric case.

One of the main problems of traditional ridge-line extraction is that it is an entirely local process. Each pixel is classified independently leading to numerical instabilities and structural inconsistencies. For example, the close-up on the ridge-lines shows that ridges are multiple pixel wide and can degenerate into ridge regions which violates the definition as well as the intuition of a "ridge-line" (Figure 12(c)). At the same time, the color map suggests that ridges are related to ascending lines of the MS complex. While there does not yet exist a mathematical theory linking these structures, experiments seem to support this notion (Figure 13).

Initially, the ridge lines seem to be a subset of the ascending lines in the MS complex, and simply filtering all unnecessary lines would provide a segmentation highly similar to the traditional approach. By definition, the ascending manifolds cannot degenerate and they are computed in a global approach making them far less likely to be influenced by numerical instabilities in the data. Furthermore, there exists a direct generalization of ascending line for the volumetric case

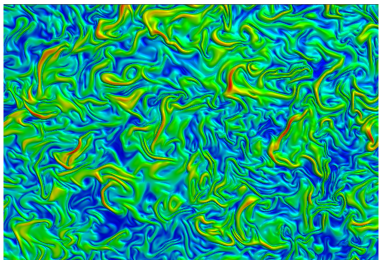


(a) Before topological cleaning.

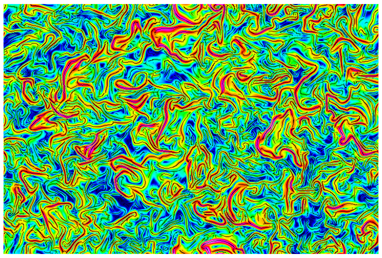


(b) After topological cleaning.

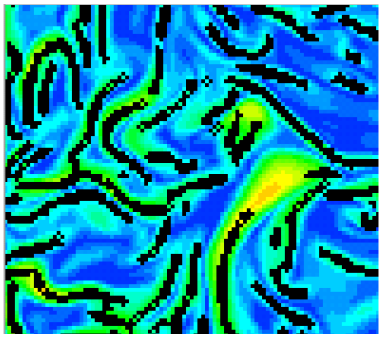
Figure 11: Comparison of raw analysis (left) and “cleaned” analysis (right).



(a) Chi field at y=448 of time step 100 drawn using a logarithmic color map.

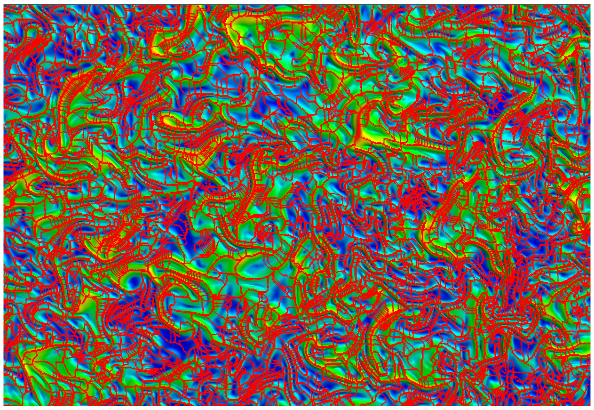


(b) Ridge lines in the chi field as extracted by traditional methods.

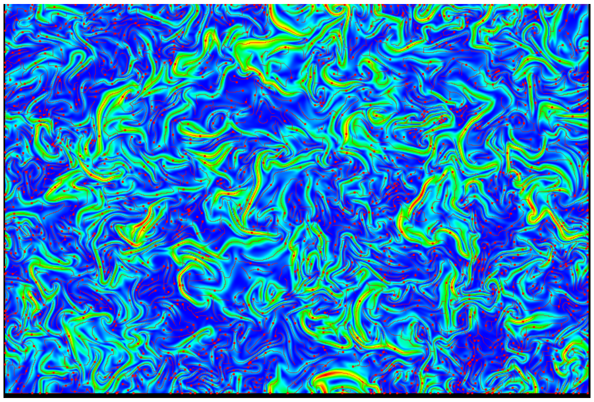


(c) Zoom-in to the ridge line segmentation showing ridges are multiple pixels thick and can become degenerate.

Figure 12: Extracting ridge lines from the chi field using traditional techniques.

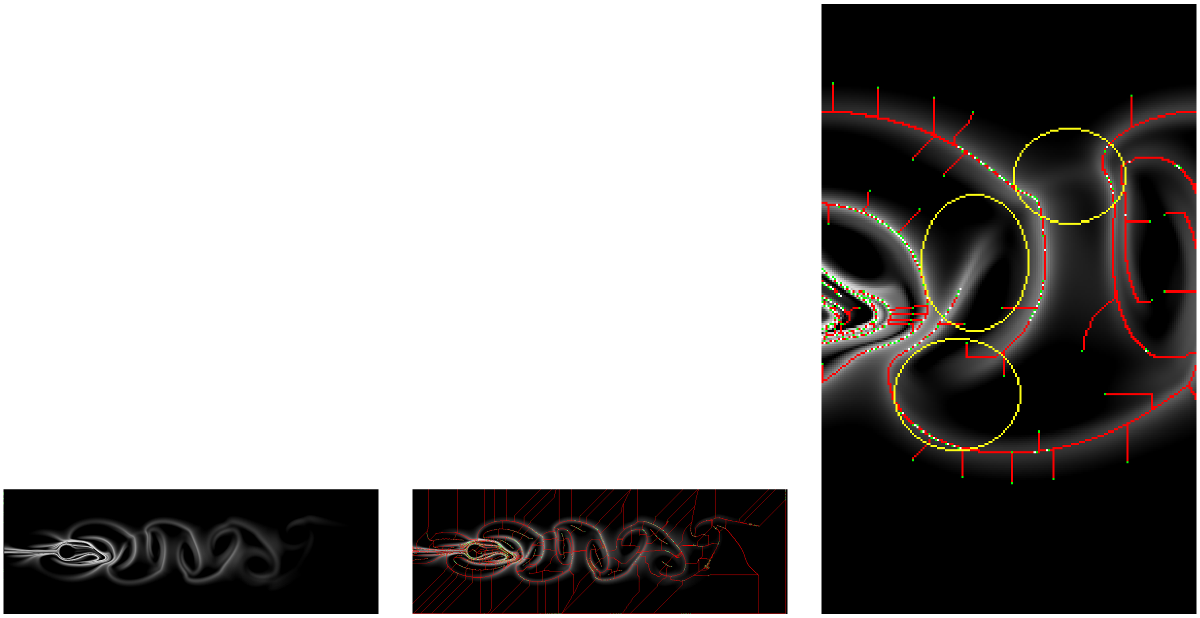


(a) All ascending lines of the MS complex of the chi field.



(b) Ascending lines of the MS complex filtered by gradient magnitude.

Figure 13: Comparison of ascending lines of the MS complex (left) with ascending lines filtered by gradient magnitude (right).



(a) An FTLE field for which we want to extract the ridges.

(b) The full set of ascending manifolds from saddles.

(c) Certain ridges are missing from this set.

Figure 14: Not every ridge is identified with an ascending manifold of the MS complex.

(a) An FTLE field modified to guarantee sufficient saddles to trace each ridge.

(b) Some ridges are still missing.

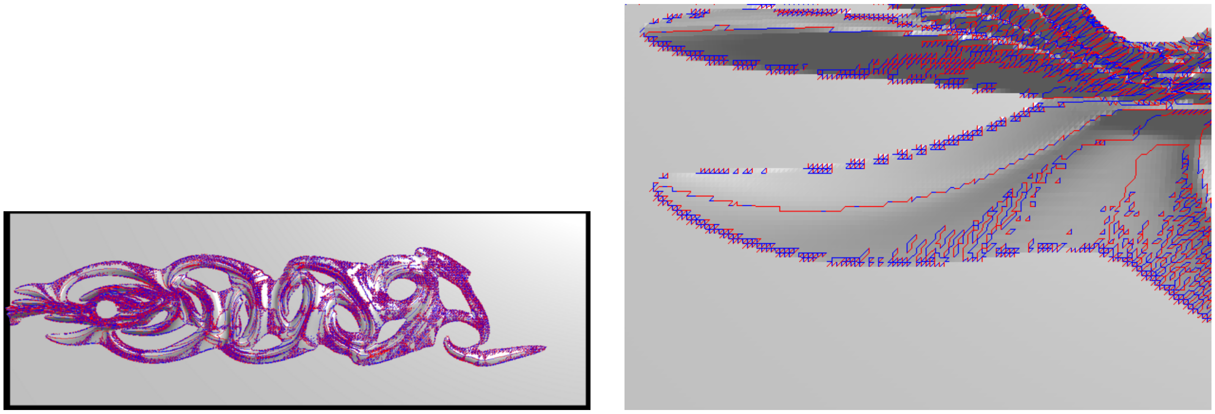
(c) Extra ridges, called "feathers" are difficult to remove.

Figure 15: Not every ridge is identified with an ascending manifold of the MS complex.

where the ridge surfaces should be described by the two-dimensional ascending manifolds in the chi-field. However, as Figure 14 illustrates, upon closer inspection, not every "ridge" is identified as an ascending manifold of the MS complex.

The fundamental problem with using the MS complex in a straightforward manner is that "ridges" are more intuitively defined on local curvature than critical point and gradient behavior. In particular, several ridges can merge together, and there will not be a saddle at the merge points to start an ascending 1-manifold for each. We investigated a technique for inserting additional saddles, by first considering the level set curvature of the "zero" value. Maxima on this curvature function are identified as saddles from which to trace ascending paths. Figure 15 illustrates that, while this identifies many more of the missing ridges, some are still missing. Furthermore, spurious ridges, called "feathers," are identified that are due completely to noise in the curvature function. The feathers prove to be extremely difficult to remove. We defined ridgy-ness criteria based on both persistence for topological simplification as well as various heuristics based on local curvature magnitude.

The missing ridges lines are due to the fact that we picked the zero level set as an arbitrary threshold for inserting extra saddles. Ridges that merge before reaching the zero level set will not



(a) Zoomed-out view of ridges. (b) Zoomed-in view shows susceptibility to noise.

Figure 16: Using a Jacobi set definition of ridges, we do not omit any ridges (left). However, this technique is particularly susceptible to noise (right).

be identified. In fact, if we generalize the notion of where to place this threshold (where to insert extra saddles to guarantee that every ridge will have an arc of the complex), then we converge on the following definition of ridge-lines: A ridge is the Jacobi set of level set curvature and the scalar function. The following figure shows that the Jacobi set computed for these two functions in fact identifies every single ridge. The major disadvantage of using Jacobi sets is that simplification is not well understood. Therefore, all the noise due to discretization of the function is represented as Jacobi set components that are difficult to remove. We will focus on defining robust criteria for simplification of Jacobi sets.

Future work for this project includes:

- Investigate new segmentation of 3D scalar dissipation structure using stable manifolds of the 3D MS complex as features.
- Deliver prototype of new ViSUS software to stakeholders.
- Integrate visualization of topological segmentation into ViSUS framework for linked views.
- Investigate the use of eigenfunctions as shape descriptors and their use in robustly measuring shape parameters such as length scales and thicknesses.
- Use particle data recently integrated into stakeholders S3D code to compute Finite Time Lyapunov Exponents and investigate the use of Lagrangian Coherent Structures to analyze flow fields.
- Investigate the use of tensor analysis for the classification of general flow fields.
- Integrate partial tracking of Jacobi sets (namely the tracks of maxima) into S3D code to facilitate more accurate tracking.

2.6 Fusion

VACET's support for the DOE fusion community is built around a combination of one-on-one relationships with specific stakeholders and conducted with, in part, a Science Application Partnership (led by A. Sanderson, Utah) with several SciDAC fusion projects. The general idea is to provide

much-needed visual data exploration and analysis capabilities to several fusion stakeholders. There are three main focus areas of our work.

Particle Path Analysis and Visualization.

Fusion researchers use simulations (particle-in-cell codes) that currently use millions to billions of particles, with each particle containing multiple scalar and vector data (aka multivariate data). They would like to have the ability to explore the nature of the particle orbits in an interactive manner. However, it is impractical to view millions or billions of particles at one time and gain insight due to excessive visual clutter. As such, the fusion scientists have expressed the need for visual data exploration and analysis tools that allow them to cull away the particles using a user defined query or other statistical means.

At the same time as the particles are displayed, scientists are seeing the particles in context of data that is not associated with the particle but is part of the simulation. This data may be scalar (electric potentials) or vector data (magnetic field) and may have its own visualization requirements. For instance, the scalar data may be viewed using a variety of techniques, such as volume rendering, slicing or iso surfaces.

Magnetic Field Analysis and Visualization

Fusion scientists are studying the affects of magnetic islands that form in burning plasma. These islands cause defects in the magnetic field and the current flow resulting in contact between previously separate regions. This contact results in hot areas coming into contact with cool areas, which leads to core cooling. Scientists have expressed the need for tools that allow them to be able to automatically generate Poincaré maps of the magnetic field and detect the island formation and track them over time.

Scientists have also expressed the desire to see other simulation variables displayed concurrent with the Poincaré maps. This additional data may be scalar (electric potentials) or vector data (magnetic fieldlines) and may have its own visualization requirements. For instance, the scalar data may be viewed using a variety of techniques, such as volume rendering, slicing or iso surfaces.

Comparative Analysis and Visualization

Fusion scientists have expressed the need for tools that allow them to perform inter- and intra-simulation comparative studies, as well as comparing experiment and simulation (synthetic diagnostics). These tools will need to be able to analyze scalar and vector data produced on different meshes and different time scales. At the same time physicists desire tools that will allow them to compare and visualize multivariate data.

Accomplishments this period.

The major emphasis has been on completing and refining the tools in VisIt needed to support Fusion analysis and visualization. This work has included additions released in VisIt versions 1.12 and 1.12.1.

During the past six months our major focus has been to enhance VisIt for magnetic fieldline analysis via the Poincaré plot and query-based visualization of particle systems (using the new FastBit query and I/O capabilities in VisIt, see Section 4.3). For the analysis of the magnetic fieldlines, we continued to improve on the infrastructure associated with the integration of streamlines. The one major infrastructure change that has greatly affected the analysis quality has been the ability to request additional integration steps on an as-needed basis. Previously, the number of integration steps was fixed. As a result, for most uses, there were either too few or too many integration steps. Consequently, it was often difficult to perform the analysis with full confidence without requesting far more (or far fewer) integrations steps than necessary, thus wasting computational resources. Now with the ability to request the number of integration steps on an as-needed basis, the fieldlines can be analyzed with greater confidence without a large increase in computational resources.

The other major enhancement in VisIt has been the deployment of the FastBit enabled query-driven visualization. Infrastructure was completed to allow range based queries utilizing FastBit and the utilization of the query results for visualization via the new VisIt "Persistent Particles" operation, which allows for the generation of both static and dynamic visualizations of particle paths. This infrastructure work required many changes through out the VisIt code and is still being hardening.

Finally, we committed the VizSchema plugin from Tech-X into the VisIt repository. We use this plugin to visualize Stephane Ethier's data.

Work targets for the future include:

- Hardening of the new tools especially the query based visualization.
- Cumulative queries.
- Additional streamline integration methods for nonlinear elements that use higher order interpolation.
- Continue working with Fusion teams and modifying tools to meet needs.
- Optimizations and improvements in streamline and analysis algorithms.
- Infrastructure to support Synthetic diagnostics.

3 Technology Incubation Projects

3.1 Streamlines

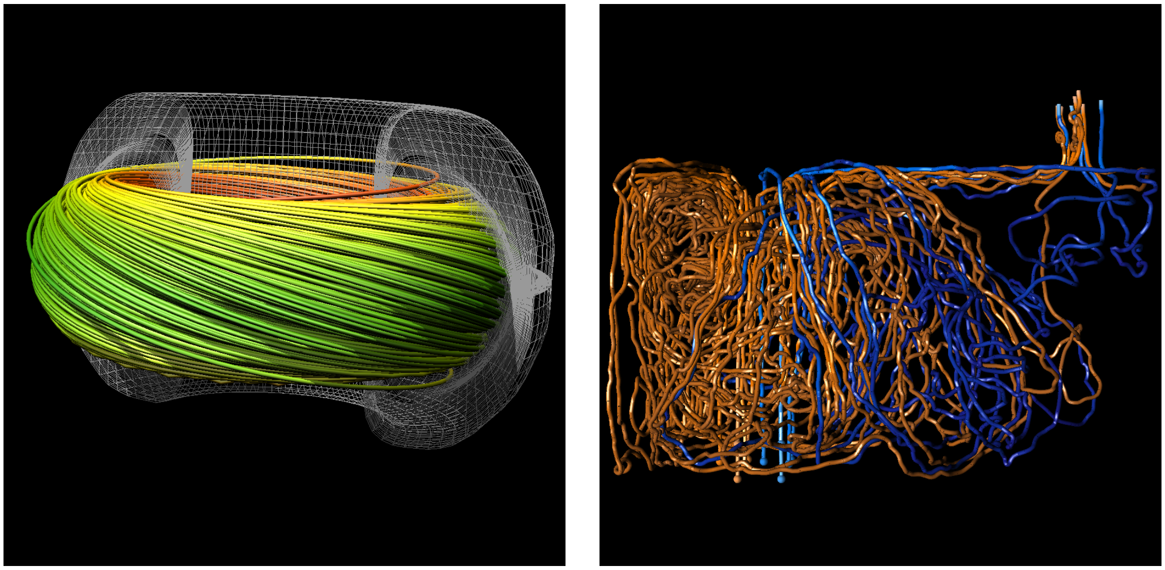
This effort has accomplished the initial goal of implementing and deploying a parallel capable streamline infrastructure with several different user definable parallelization strategies.

Future goals focus on the following areas.

- **AMR data** poses special challenges. First, the streamlines algorithm needs to correctly deal with the nesting of patches and the transition across patches of differing coarse/fine resolution. Second, streamlines need to have continuity (C0 or C1, depending upon user choice) across patch boundaries.
- **Poincaré plots** and other forms of analysis that rely on integral curve evaluation.
- **Improved understanding of parallel performance.** It is clear that there is not one best algorithm for all cases. At least three algorithms will be deployed which parallelize work over Streamlines, Domains and a hybrid that parallelizes over both streamlines and domains.
- **Pathlines**. Pathlines calculations are like streamline calculations, except the velocity field advances in time as the particle is advected.
- **Additional algorithms** like Initial Value Problem (IVP) solver types with advanced features (e.g., continuous output).

Accomplishments this reporting period:

Parallel streamlines. We completed design and implementation of a hybrid parallelization algorithm, integrated it into VisIt, and have a technical publication about this work that is part



(a) Streamlines showing the magnetic field in NIMROD fusion simulation output.

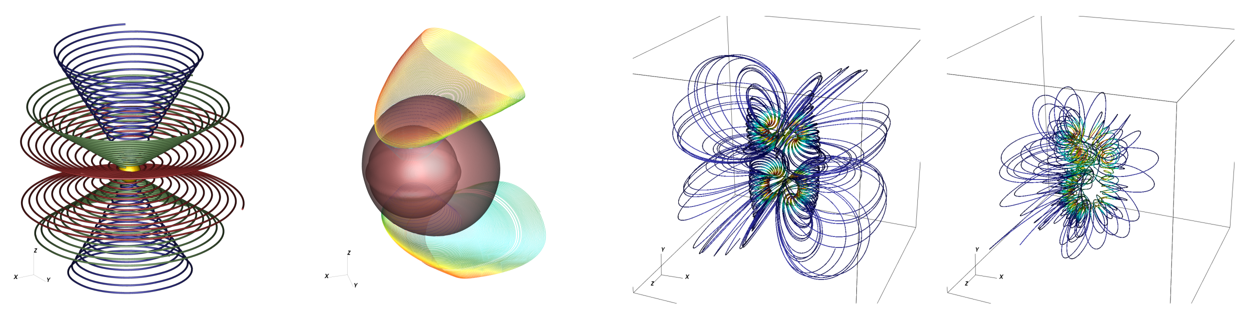
(b) Streamlines showing fluid flow in data produced by the NEK3D thermal hydraulics simulation.

Figure 17: These images show visual output from our new parallel streamlines algorithm applied to simulation data from two different application domains.

of the SC09 technical program. Figure 17 shows example output from our new parallel streamlines algorithm.

AMR Streamlines. We implemented a preliminary streamline integration algorithm for block-structured AMR data. To better understand streamline quality, we compared several interpolation schemes (Dual-Mesh, averaging to mesh nodes, nearest neighbor) in order to integrate in cell-centered data sets. To explore options for addressing the problems caused by patch boundaries in AMR data sets, we explored explicit treatment of resolution level boundaries as well as boundaries between different domains in the same level, and applied the method to several AMR simulation data sets. We presented results of this new method at the 2009 Dagstuhl workshop and have a technical paper in submission. Figure 18 show example output from the preliminary AMR streamline algorithm.

Work targets for the future for streamlines, AMR streamlines, and pathlines include: (1) ex-



(a) Magnetic field of the sun: Parker spiral shaped field lines.

(b) Magnetic field of the sun: field lines pass the termination shock.

(c) Two merging vortex cores, $t = 0$.

(d) Two merging vortex cores, $t = 60$.

Figure 18: Example output from our preliminary AMR streamlines algorithm.

ploring optimizations in implementation and parallelization beyond current methods as well as more efficient hybrid parallel algorithms; (2) software engineering to include our AMR streamlines algorithm into VisIt; (3) explore application of an algorithm extension to incorporate Embedded Boundary/Material interface data sets.

3.2 Embedded Boundaries/Material Interface Reconstruction

Many important classes of computer simulations of physical phenomena require support for materials, i.e. discrete regions of space with different physical properties. For example, a simulation of tidal waves needs to partition space into water and air, and a simulation of an automobile accident must model glass, metal, and rubber. There are two approaches to supporting materials on the computation mesh: Lagrangian (where each cell contains exactly one material for the entire simulation) and Eulerian (where the materials are allowed to flow through the mesh). Although the Lagrangian approach is simpler to implement, the Eulerian approach is often used because of its flexibility. The Eulerian approach is ideal for computations requiring a static mesh while materials move, for materials that bend and twist so significantly that they cant be represented easily with normal mesh elements, or simply to model materials at a higher resolution than the mesh to maintain accuracy. The result is that cells in the computational mesh will be mixed, i.e. containing two or more materials.

Since the computational mesh also is the native storage mechanism for simulation data, methods for keeping track of this material information are not straightforward. A common approach is to use volume fractions (VF), to store in each cell the percent of the cell occupied which each material.

There are multiple correct solutions and many criteria for a good reconstruction: Does it honor the volume fractions? Does it place materials from neighboring cells next to each other? Does it create large discontinuities? Although simulations reconstruct interfaces themselves, their primary concern is advecting materials through the mesh correctly, not visualization and analysis. Their reconstructions often lead to inaccurate analysis and poor aesthetics.

For this effort, we are pursuing algorithms that are well suited for visualization and analysis. Our primary EB customer at this point is the SciDAC Applied Partial Differential Equations Center (APDEC) and the science projects they support. Future potential customers include the Center for Nanoscale Control of Geologic CO_2, one of DOE's Energy Research Frontier Centers. We work closely with APDEC during our research to ensure that our new algorithms meet their science and analysis needs.

Progress this reporting period:

- Committed a new revision of a material interface reconstruction algorithm into the main VisIt source code trunk; it will appear in a future public release.

- Integrated into the VisIt trunk a Youngs-style interface reconstruction algorithm. This algorithm produces an excellent reproduction of the volume fraction but the resulting surface has poor connectivity.

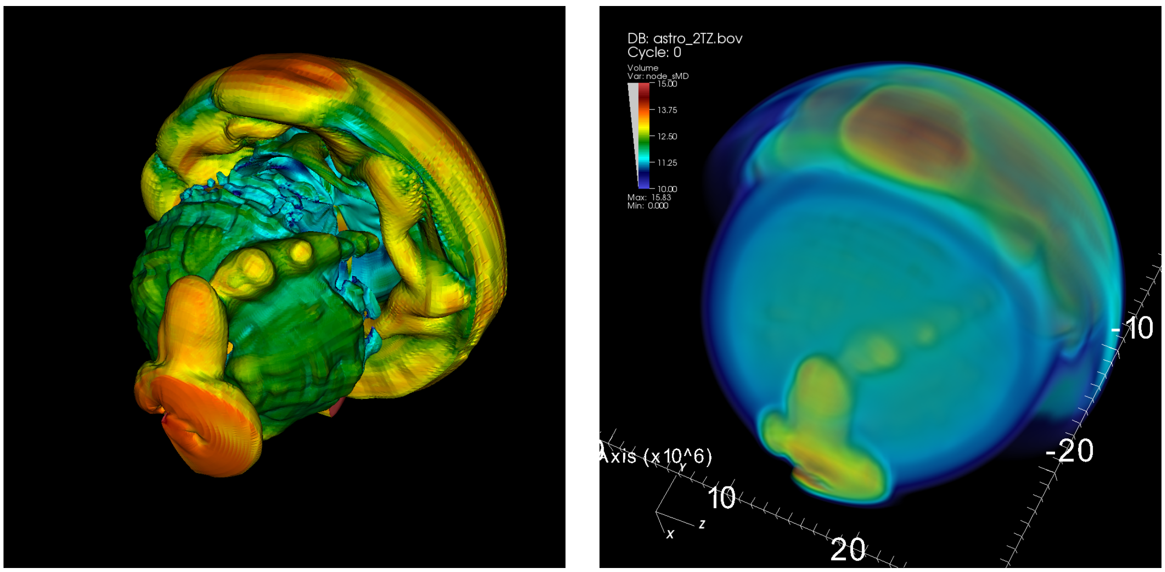
- Integration of an alternative algorithm into VisIt – created by VACET researchers – for testing and evaluation. This algorithm can produce highly accurate results, but at a significant cost in terms of time/space requirements. It is not yet clear if this algorithm will be "the winner" in the long run due to this accuracy/speed tradeoff.

4 Common Infrastructure Projects

4.1 VisIt Hero Runs

4.1.1 Background

As the by-product of advances in technology is “more and more data,” one issue facing the visualization and analysis community is the feasibility of using today’s largest computational platforms for knowledge discovery. To gain better insight into this issue, VACET researchers recently conducted a series of experiments aimed at fostering a better understanding of functional and performance limits that might be encountered when running a production-quality visualization application at extreme levels of concurrency on data sets of unprecedented size. The results, which we discuss in this section, suggest this approach is viable and that visualization research and development efforts have produced technology that is today capable of ingesting and processing tomorrow’s data sets.



(a) Isocontouring of two trillion zones on 32,000 Opteron cores of JaguarPF, a Cray XT5 at OLCF/ORNL.

(b) Volume rendering of two trillion zones on 32,000 Opteron cores of Franklin, a Cray XT4 at NERSC/LBNL.

Figure 19: Our functional performance experiments consist of loading extremely large data sets and executing visualization algorithms at extreme levels of concurrency producing images of isocontouring (left) and volume rendering (right).

Another purpose of these runs was to prepare for establishing VisIt’s credentials as a “Joule code,” or a code that has demonstrated scalability at a large number of cores. VisIt is the first and only visual data analysis code that is part of the ASCR Joule metric, which aims to track code performance (scalability) over a period of time.

The team’s experiments consisted of running the VisIt software application on several of the nation’s largest computing platforms and on data set sizes ranging from 500 billion (two terabytes per scalar) to 2 trillion cells (eight terabytes per scalar) and at concurrency levels ranging from 8000 to 64,000 cores. Each experiment consisted of running VisIt in parallel: loading in data, performing two common visualization tasks (isosurfacing and volume rendering), and producing an image (Figure 19).

4.1.2 Parallel Processing of Data

As there is some variation in how tools process data in parallel, we will describe VisIt's approach for parallel processing. VisIt employs a client-server model. The VisIt server is parallelized using an MPI-based communication model and can run on a large, parallel platform. The VisIt client, which runs on the user's workstation, communicates with the VisIt server over an IP-based connection. To begin, the end user sets up the visualization or analysis task from the VisIt client-side interface. The client then communicates this information to MPI rank 0 of the server. MPI rank 0, in turn, communicates this information to the other processors through a broadcast command. Each processor then sets up a data flow network, which consists of modules to read data, process data, and ultimately to render data to images. VisIt's parallelization approach uses domain decomposition so that each MPI process, which runs data I/O, processing, and rendering, operates only on its data subset. The partitioning occurs by operating on meta-data, for example by partitioning pre-defined domains.

Once the data flow networks are set up on each MPI process, execution begins. First, each process sets up its reader to load only the data for its subset of the larger data set. Next, each process executes one or more algorithms (e.g., isocontouring) with user-specified parameters (e.g., isocontouring level). For processing sequences that includes rendering, each processor renders its portion of the problem into an image, then the resulting images from all processes are assembled into a final image. The resulting image is then transferred to the client and displayed to the user. The processing and rendering for volume rendering is somewhat different; the volume rendering algorithm is not embarrassingly parallel.

4.1.3 Experiment Description

One goal of this experiment was to demonstrate the viability of these techniques across diverse supercomputing environments, in terms of operating system, I/O performance, FLOPs, and network bandwidth. We performed these tests on Crays (OLCF/ORNL's Jaguar & NERSC/LBNL's Franklin), a Sun Linux machine (TACC's Ranger), a CHAOS Linux machine (LLNL's Juno), an IBM AIX machine (LLNL's Purple), and an IBM BG/P machine (LLNL's Dawn).

Because each machine has a different number of cores, we performed a weak scaling study. We started with a single data set, which we then upsampled to an appropriate resolution. This data set was from a core-collapse supernova simulation done by the CHIMERA code, on a curvilinear mesh of more than three and one half million cells[3]. We chose the upsampling approach since we are not aware of any current data sets containing 2T cells and since the primary objective for our studies is to better understand the performance and functional limits of parallel visual data analysis software. These objectives can be achieved using upscaled data.

The upsampling process involved interpolating a scalar field onto a high resolution rectilinear mesh and then writing the data out as compressed binary data (gzipped). There were ten files for every core used, and every file contained 6.25 million data points, for a total of 62.5 million data points per core. In our experience, the visualization tool often has one tenth (or less) of the resources (e.g., cores) as the simulation code and simulations codes often write out one file per core. Hence, having multiple files per core was our best approximation at emulating these common real world conditions.

We ran with 16000 cores on each machine visualizing one trillion cells[4]. On the Jaguar and Franklin machines, we ran additional, larger-sized problems consisting of two trillion cells on 32000

[3] Sample data courtesy of Tony Mezzacappa (ORNL), Bronson Messer (ORNL), Steve Bruenn (Florida Atlantic University) and Reuben Budjiara (University of Tennessee).

[4] We ran with only 8000 cores and one half trillion cells on Purple, because the full machine has only 12208 cores,

cores, and four trillion zones on 64000 cores on Dawn (a BG/P machine at LLNL). Although times varied from machine to machine, I/O was the dominant factor (taking two or more minutes at 16000 cores), with contouring taking approximately ten seconds and rendering taking one to ten seconds.

4.1.4 Issues Discovered During Scaling Study

Although the majority of VisIt's infrastructure scaled well to a large number of cores, we ran into several obstacles along the way:

- VisIt's MPI rank 0 was collecting status information from all other processors through point-to-point communications, which is non-scalable. For our study, we were forced to work around this issue. Soon thereafter, Mark Miller of Lawrence Livermore independently encountered this problem and added a fix.
- VisIt's volume rendering algorithm was attempting an optimization for sample point communication that required a buffer that had an $O(nProcs \times nProcs)$ space requirement. This "optimization," while appropriate for low levels of concurrency, proved to cause problems at high levels of concurrency. The fix, implemented by VACET and available on the mainline version of VisIt, is to eschew this optimization at high levels of concurrency.
- We observed that the loading of shared libraries took quite a long time at scale (as much as five minutes) and VisIt's plugin model may need to adapt for this case, likely by switching to a static binary with precompiled plugins. To address this, we added an option to VisIt's build system to perform static linking. We will evaluate the extent of the performance improvement soon.

4.1.5 Lessons Learned

The primary objective for these experiments was to gain a better understanding of functional and performance limits when running visual data analysis applications at extreme levels of concurrency and and problem sizes. We encountered a couple of minor problems that will be rectified and appear in a future public VisIt release.

From a visual data analysis perspective, these problem sizes and concurrency levels are a "first." The successful completion of these functional and performance tests show progress towards petascale computing by demonstrating that today's technology is capable of ingesting and processing tomorrow's data sets.

The performance data the team collected during the experiments reveals insights into potential bottlenecks and opportunities for performance optimization on different machine architectures at high levels of concurrency and ultrascale data sets. Future work will include a more detailed, end-to-end performance study of several different visualization algorithms to better understand performance limits and opportunities for VisIt, a production-quality visual data analysis software application.

4.1.6 Publications and Press

- H. Childs, D. Pugmire, S. Ahern, B. Whitlock, M. Howison, Prabhat, G. Weber, E. W. Bethel. *Extreme Scaling of Production Visualization Software on Diverse Architectures.* Accepted

and only 8000 are easily obtainable for large jobs.

for publication in IEEE Computer Graphics and Applications upcoming special issue on Ultrascale Visualization, to appear 2010. This article contains more information about the experimental methodology, results, and "speed bumps" encountered along the way.

- E. W. Bethel, C. Johnson, S. Ahern, J. Bell, P.-T. Bremer, H. Childs, E. Cormier-Michel, M. Day, E. Deines, T. Fogal, C. Garth, C. G. R. Geddes, H. Hagen, B. Hamann, C. Hansen, J. Jacobsen, K. Joy, J. Krger, J. Meredith, P. Messmer, G. Ostrouchov, V. Pascucci, K. Potter, Prabhat, D. Pugmire, O. Rbel, A. Sanderson, C. Silva, D. Ushizima, G. Weber, B. Whitlock, K. Wu. "Occam's Razor and Petascale Visual Data Analysis." In Journal of Physics Conference Series, Proceedings of SciDAC 2009. LBNL-2210E.

- Scientific Computing article on trillion-zone runs. August 27, 2009[5].

- VizWorld article speculating on the importance/impact of the trillion-zone run articles. August 27, 2009 [6].

- HPCwire article on trillion-zone runs. June 10, 2009 [7].

- VizWorld article on trillion-zone runs [8].

4.2 VisIt Passes Joule Software Metric

The Joule program within DOE ASCR is an important metric to ensure that codes seen as important to the DOE ASCR's mission are making effective and efficient use of HPC resources. For 2009, VisIt was chosen as a Joule program candidate. In October of 2009, VisIt passed the Joule metric and is now an official Joule code. VisIt is the first ever non-simulation code to be selected for Joule code certification.

The Office of Management and Budget (OMB) uses its Performance Assessment Rating Tool (PART) to perform evaluations of different programs. PART has seven worksheets for seven types of agency functions, which includes R&D activities. In FY 2003, the DOE Office of Science worked directly with the OMB to come to a consensus on an appropriate set of performance metrics consistent with PART requirements. The scientific performance expectations of these requirements reach the scope of work conducted at the national laboratories. The *Joule* system emerged from this interaction. Joule enables the CFO and senior DOE management to track annual performance on a quarterly basis.

In FY09, OASCR's Joule goals include two primary performance metrics. The first concerns job size (e.g., level of concurrency) at NERSC. The second, and of interest in this discussion, concerns improving average performance increases in computational effectiveness (primarily through weak scaling studies).

In FY09, the VisIt visualization system is part of the Joule metric. VACET researchers from Oak Ridge National Laboratory are leading this effort. VisIt is the only visualization application that is included in the FY09 Joule metric.

For the Q2 benchmark, we have selected two common analysis and visualization techniques, isosurface extraction and volume rendering. These two algorithms were run on the output of a recent simulation of the Denovo radiation transport code on the dose concentrations around

[5] `http://www.scientificcomputing.com/article-hpc-Smashing-the-Trillion-Zone-Barrier-082709.aspx`.

[6] `http://www.vizworld.com/2009/08/smashing-the-trillion-zone-barrier/`.

[7] `http://www.hpcwire.com/topic/visualization/DOE-Researchers-Test-Limits-of-Visualization-Tool-47533672.html?viewAll=y`.

[8] `http://www.vizworld.com/2009/06/doe-researchers-push-visit-to-the-limit/`.

a reactor core in a nuclear power generating plant. These runs demonstrate scaling in several ways. The first is weak scaling in extracting several different dose contours from the energy groups computed by the simulation code. Second, we demonstrated weak scaling in two aspects of the volume rendering: scaling with respect to the problem size, and scaling with respect to the number of samples computed along each ray.

In post-production analysis and visualization tools, the most important benchmarking metric is the time it takes to render a frame to a user display. Under normal uses cases, the user loads the simulation data from disk into memory and then repeatedly interacts with the data, successively modifying isovalues and observing the results. We therefore focused on the scalability of the isosurface extraction algorithms and ignored the scalability of the onetime expense of reading simulation data from disk. As in isosurface extraction, the time to render frames once the data is loaded from disk is the most relevant metric to end-users. We therefore focused on the scalability of the volume rendering algorithms and ignore the scalability of the one-time expense of reading simulation data from disk.

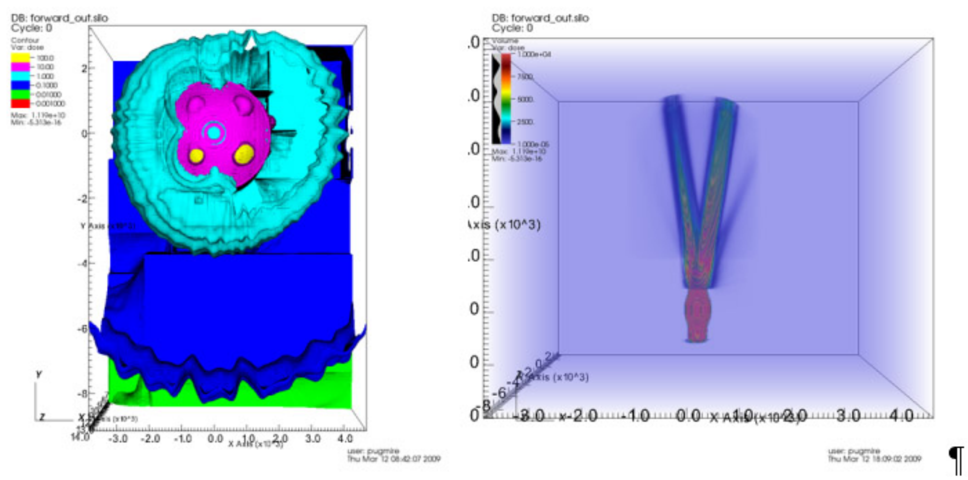


Figure 20: Extraction of radiation dose contours (left) and a volume rendering (right) from the nuclear power plant simulation results output from the Denovo radiation transport code.

The output produced from the VisIt analysis and simulation runs produced expected results that have been verified with the code developer and are deemed acceptable. We have therefore accepted the results for the Q2 benchmark runs.

Summary of activities this period:

- March 2009. Successfully ran baseline calculation on a Denovo (radiation transport code) simulation on 4096 cores of jaguarpf.
- June 2009 Improvements to VisIt to enhance scalability:
  - The isosurfacing algorithms scaled very nicely, as expected.
  - In the Q2 runs, a parallel deficiency in the volume rendering code was discovered. When running the baseline calculations out to 4096 cores, it was discovered that scalability dramatically dropped off after 1024 cores. This was a new and important discovery as the algorithms had never been formally studied at this scale. This resulted in two important updates being made to the volume rendering algorithms:

    * First, parallel volume rendering is a communication-expensive task. To minimize communication, there is an optimization step that occurs to determine the distribution of data and decide how best to allocate it to the computation resources. It was discovered that this optimization step does perform well on large number of processors. This bottleneck was identified and fixed, resulting in a 4.5X performance increase in volume rendering.
    * Second, parallel volume rendering can also be a very memory expensive task. To optimize memory use, the algorithm can take advantage of the fact that ray-casted volume rendering is an image space algorithm, and therefore can be broken up into independent image tiles. This has the advantage of using less memory at the expense of some parallel inefficiency. This tradeoff is acceptable at lower levels at parallelism where memory is at a premium. However, at larger scales, where memory is less at a premium, this tradeoff can become a bottleneck to scalable performance. Again, by running beyond previous levels of parallelism, this bottleneck was identified and fixed.

- September 2009. Re-ran code on the benchmark problem and showed weak scaling.
  - Isosurfacing algorithm showed slightly super-linear weak scaling, as was expected.
  - Improvements to volume rendering code resulted in a 4.5X performance increase.

4.3 Integrating FastBit and VisIt

The main objective for this project is to develop methods for fast visual exploration of extremely large data. The goal is to enable the stakeholders to interactively explore their data, with a current focus on particle data. All applications (accelerator and fusion) require efficient means for identification of relevant particle subsets (e.g., particle beams) as well as efficient means for tracing of particles over time.

Accomplishments this period:

- Integrated FastBit/HDF5-FastQuery with VisIt via a new HDF_UC file reader. The HDF_UC project, from the LBNL visualization research effort, provides the fundamental building blocks for integrating FastBit and the HDF5 file format/API.

- Developed means for efficient rendering of histogram-based parallel coordinates in VisIt. These serve as main interface for performing data selection.

- Accelerate threshold and id-based queries by evaluating the queries directly in the file-reader using FastBit.

- Accelerate particle tracing by using a series of ID-based queries evaluated by the file-reader. Implemented new PersistentParticles operator.

- Added concept of NamedSelections to VisIt to support persistent selection of data objects based on ID to support tracing of particle subsets over time.

- Studied performance of 2D histogram computation and ID-queries.

- Deployed functionality in VisIt 1.12.1.

Future work for this effort includes:

- Parallel implementation of particle tracing in VisIt.

- Named Selections: (1) Develop GUI for management of Named Selections; (2) Support for cumulative queries. Examples of cumulative queries can be simple combinations of per-time step queries – e.g, find all particles that ever exceed a threshold of $px > 1e10$ – as well as more complex queries that require comparison of different time steps.

- Integrate indexing with VisIt convert to support bitmap index creation directly using VisIt.

- Develop visualizations for investigation of analysis results: develop VisIt macros for standard plots used for the investigation of analysis results.

- Investigate further applications.

- Extend FastBit+VisIt to support QDV for grid-based data (e.g., field data in the context of the accelerator data).

- Comparative analysis of particle simulations: develop methods to compare simulations based on user-defined particle beams.

4.4 Tuvok

VACET researchers have developed cutting-edge techniques for visualizing large volume data. Through an emphasis on real-time performance, scientists are encouraged to explore their data and develop new theories. To achieve this level of performance, the Tuvok volume rendering library leverages the immense computational resources available on recent graphics processing units, and will be paired with more traditional parallel computing resources. Ongoing work is underway to integrate these tools into production-quality general-purpose visualization and analysis tools, such as the VisIt software package.

A common way to gain insight into volumetric data is through a technique known as "volume rendering." Due to its ability to efficiently communicate three dimensional structures, volume rendering is applicable to a wide variety of disciplines. Due to its inherently three dimensional nature, a volume renderer must examine a large amount of data before displaying an image. As simulation data grow in size, traditional volume rendering applications are strained under the increased workload. To combat this issue, scientists from VACET have developed Tuvok, a volume rendering library that harnesses the immense computational capabilities of commodity graphics processing units (GPUs). Using Tuvok, scientists can use a typical workstation rather than a small supercomputer to visualize the 8 gigabyte Richtmyer-Meshkov data set (Figure 21.) With Tuvok's level-of-detail features enabled, the data set can be interacted with in real time, at rates exceeding 60hz.

One of VACET's mission objectives is to bridge the gap between field-leading research and deployment in production-quality software. To that end, this project aims to perform the software and release engineering needed to deploy Tuvok into VisIt. In addition to the expected performance gains already discussed, this new technology produces noticeably superior visual output than existing techniques (see Figure 4.4.)

Accomplishments during this reporting period include:

- Tuvok 1.0 released, download-able from `https://code.sci.utah.edu/gf/project/Tuvok/`.

- Proof of concept serial integration into VisIt.

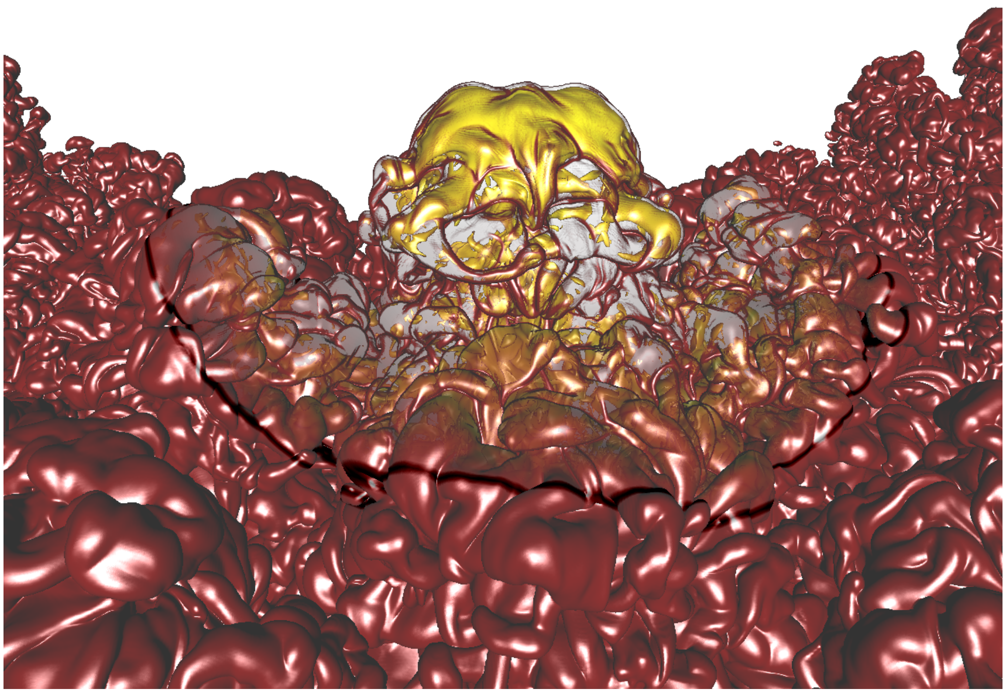
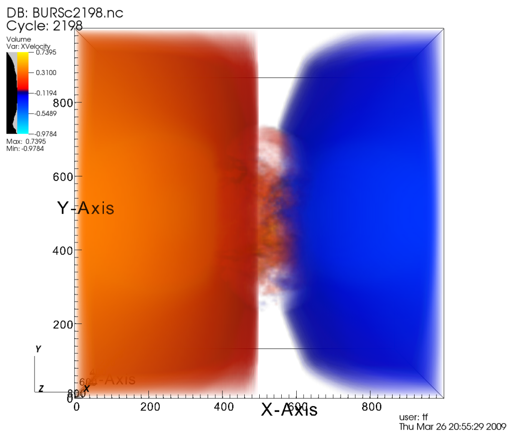
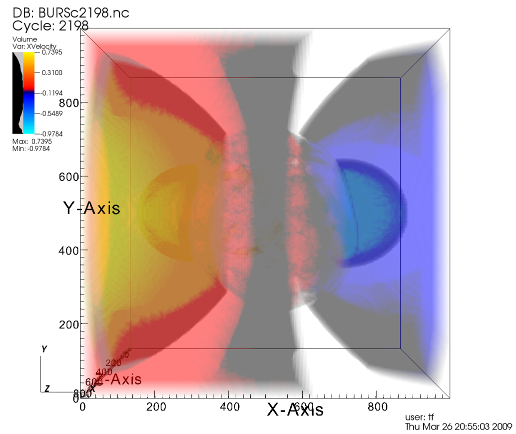


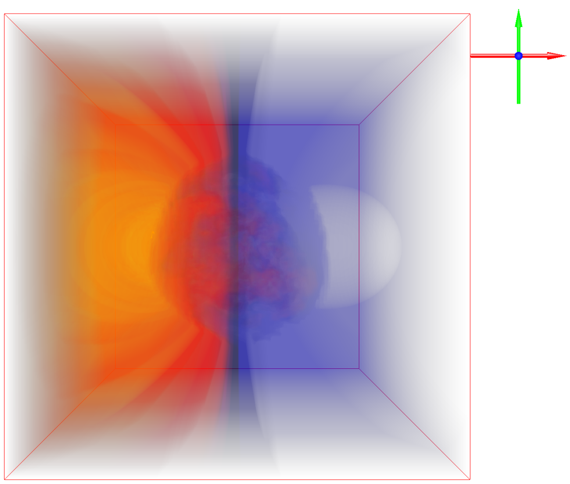
Figure 21: Tuvok using its “ClearView” rendering mode. ClearView suits the focus-and-context exploration model, common in large-scale scientific visualization, through an intuitive model of a lens which can see 'in' to the data set. In this data, scientists can rapidly discern the internal behavior of a Richtmyer-Meshkov simulation at the interface, while simultaneously viewing the global structure of the fluids.



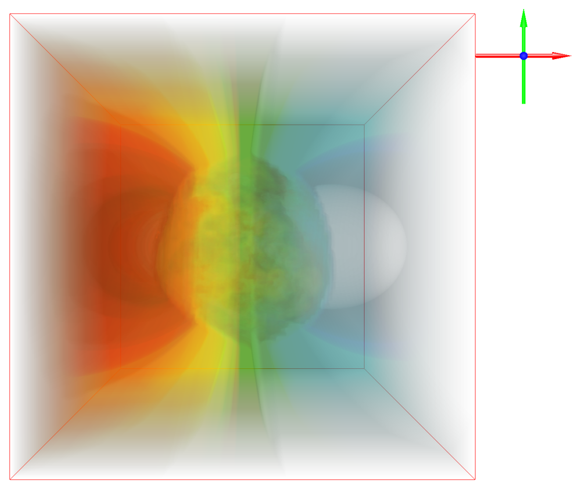
(a) VisIt's 3D texturing based volume renderer.



(b) VisIt's SLIVR volume renderer. renderer.



(c) Serial Tuvok in VisIt.



(d) Serial Tuvok in VisIt with a better transfer function.

Figure 22: Comparison of VisIt's existing volume rendering technology (top row) and output from the Tuvok engine (bottom row).

- New infrastructure to allow Tuvok to work in parallel:
  - GLEW: loading GL implementation at runtime.
  - Mesa upgrade: allows Tuvok to render via software.

  Figure showing comparisons of old and new Tuvok-based volume rendering.

  Work targets for the next six months include:

  - Bug fixes for core VisIt that effect Tuvok, e.g. camera. Targeting for public release in early 2010.
  - Stabilization of GLEW and Mesa changes. Targeting for public release in 2010.
  - Generate a technical publication about this work in 2010.

4.5 VisIt Parallel Rendering and Compositing

VisIt sees a lot of use in high end supercomputing environments, because it is one of the few tools available for visualization of extremely large data sets. When rendering data at this scale, it can happen that the amount of geometry generated during data processing exceeds what can reasonably be processed by even a high-end GPU. Moreover, many users run VisIt in client server mode, where the data live and are processed on a remote resource, and visualizations are forwarded to a lightweight client. Even if the client GPU is capable of rendering the generated geometry in real time, the overhead of sending such data over the network can be prohibitive.

To combat these problems, VisIt provides a "scalable rendering" mode (SR) that moves the rendering on to the server side. This occurs in parallel and is composited together over MPI. The resulting image is forwarded back to the lightweight client, optionally compressed, potentially saving large amounts of network bandwidth.

The VisIt team is always looking for opportunities to improve the responsiveness of VisIt in this use case. One way we do this is by integrating parallel compositing libraries, like the Image Composition Engine for Tiles (IceT[9]) into VisIt proper.

During this period, the VACET team has fully integrated IceT into VisIt. It is now the default mode by which the VisIt team verifies parallel rendering in its nightly regression tests. Example output from this work is shown in Figure 23.

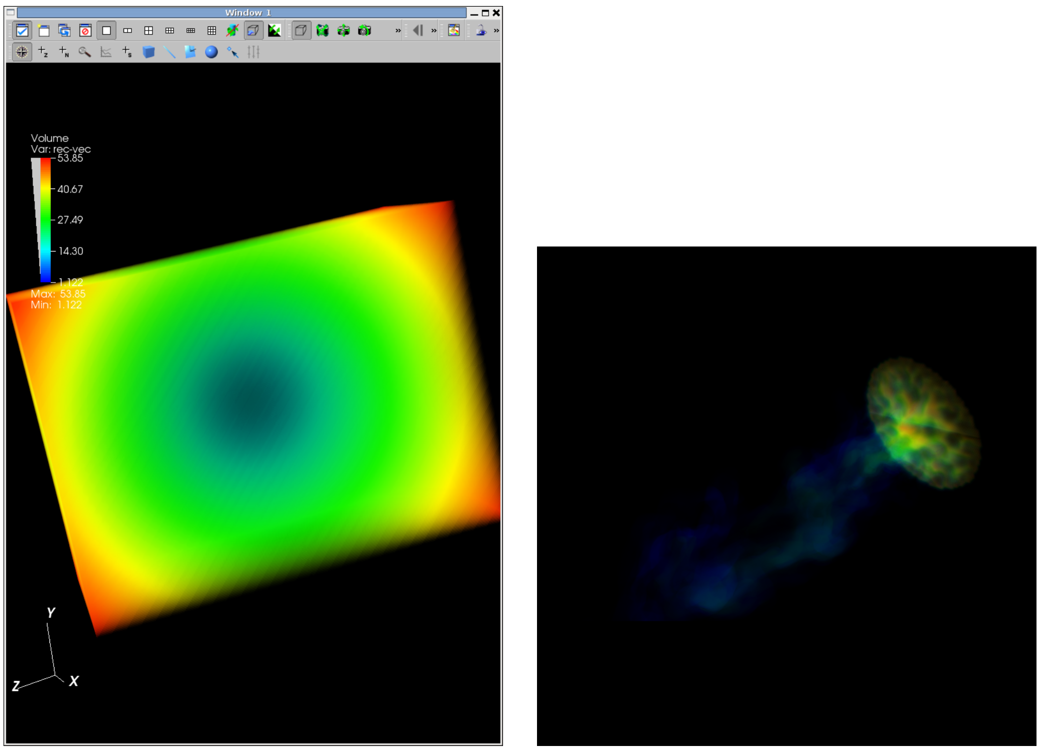
Future work includes testing to better understand the limits of this approach on larger distributed memory GPU systems/configurations and with larger data and image sizes; scalability assessments.

4.6 Data Parallel Analysis and Graphics with R

The high level objective for this work is to perform the software development and engineering work necessary to enable use of parallel computing infrastructure for use on data analysis. To that end, this work focuses on working with science stakeholders to perform data analysis work in conjunction with software engineering on a version of the R statistical package that runs on distributed memory parallel machines.

There were two issues driving progress in this reporting period. First, some of the higher level functions of Rmpi were found to be difficult to port to other platforms and exhibited unstable behavior on the original Lens analysis system. Second, we did not have a large and scalable data-parallel analysis code that could be used to drive the development and porting to other

[9] See `http://www.cs.unm.edu/~kmorel/IceT/index.html`.



(a) Tuvok rendering a test data set. Half the data are rendered on an NVIDIA GPU, the other half rendered in software via Mesa. Both halves are composited together via the new IceT infrastructure in VisIt then presented to the viewer.

(b) Tuvok rendering a helium data set on Lens, a GPU cluster at ORNL. The image was generated using volume rendering on two GPUs and two CPUs with source data split evenly across the four renderers.

Figure 23: Example output in VisIt of the new IceT compositing infrastructure.

systems. As a result, progress this reporting period has centered around the development of a scalable demonstration code that uses the strengths of R while using only the basic (and stable) functionality of Rmpi.

We developed a code for data-parallel k-means clustering that includes cluster uncertainty. The important feature of this code is that it computes cluster uncertainty based on the level of agreement between random starts. This computation accounts for expected agreement in a way that is appropriate for highly unequal cluster sizes and massive data sets. Uncertainty is the driver for determining the appropriate number of clusters for a given data set. The code is also expected to have good scalability properties due to the added burden of uncertainty computation and also due to a potential generalization to other model based clustering methods that are more computationally demanding than k-means.

The code has been tested on up to 256 cores of lens on a 3 GB climate data set that has 120 million five-dimensional data points. It takes about 40 minutes of wall clock time on 256 cores to read the data, to compute, and to evaluate a given number of clusters. The variability in this data set appears to support up to about six clusters. Six random starts begin to display considerable disagreement beyond six clusters.

In the future, our work targets are: develop documentation for running interactive and batch data-parallel R codes; port data-parallel batch R capability to ORNL's Jaguar system; collaborate with other VACET researchers and scientific stakeholders to expand porting operations to additional platforms. In the longer term, our aim is to develop more generic NetCDF and HDF readers that can be more easily adapted to other applications, and to provide more portability for the data-parallel R environment and develop more collaborations for its use.

4.7 VisTrails and VisIt

We created a VisTrails plugin for VisIt to add the provenance tracking capability to the system, which allows VisIt's users/scientists to keep track all of their simulations, data analysis, and rendering pipelines. This can make the exploration process more efficient since users can easily go back any point in time with their experiments and quickly modify the pipelines with new parameters, operators.

We have a first version of our plugin working on the current VisIt's trunk (eventual 2.0 branch). Some modifications to the actual VisIt had to be made but they are minor and more likely feature-requests to VisIt. We were able to build a binary for VisIt+VisTrails for Linux and Windows.

The high level milestones for this project are:

- A1. Integrate VisTrails Python with VisIt (completed).
- A2. Monitor changes (events occurred) to VisIt (completed).
- A3. Serialization of States in VisIt (completed).
- A4. Create customized VisTrails actions and diff mechanism for VisIt (completed).
- A5. Build VisIt/VisTrails binary for deployment (ongoing).
  - A5.1 Compilation of VisIt trunk on Windows (currently no support in trunk yet) (completed).
  - A5.2 Merge VisIt modifications back to trunk for release (ongoing).

A1. We currently use the VisIt CLI interface to load VisTrails PyQt app. This loads VisTrails GUI and takes over the main event loop of VisIt (while still dispatching events to VisIt).

A2. In order to monitor changes to VisIt during user interactions, we use the underlying callback system of VisIt. However, there was no notification if an action/interaction is done or not, we have to add this feature into VisIt.

A3. VisIt's new capability of import/export session files is used for serialization current states of VisIt. This is a lightweight, only meta-data format which can be loaded/stored efficiently.

A4. VisIt session files are not built incrementally, so a "diff" has to be performed every time a session is serialized. This helps VisTrails to keep only minimal changes of the pipeline. Compression is also used to reduce storage space.

A5.1. Building VisIt on Linux is straight forward, however, it is not the case on Windows. This is mostly because VisIt's trunk and the dependency libraries such as Mesa, GLEW and VTK codes are not up-to-date on Windows. Modifications has to be made to project files as well as Windows branch to reflect to the Linux build. Figure 24 shows the screen shots of VisIt in Linux and Windows.

Future work will consist of: merging VisIt modifications needed to support the VisTrails plugin back into the main VisIt source code tree; test using remote server infrastructure of VisIt.

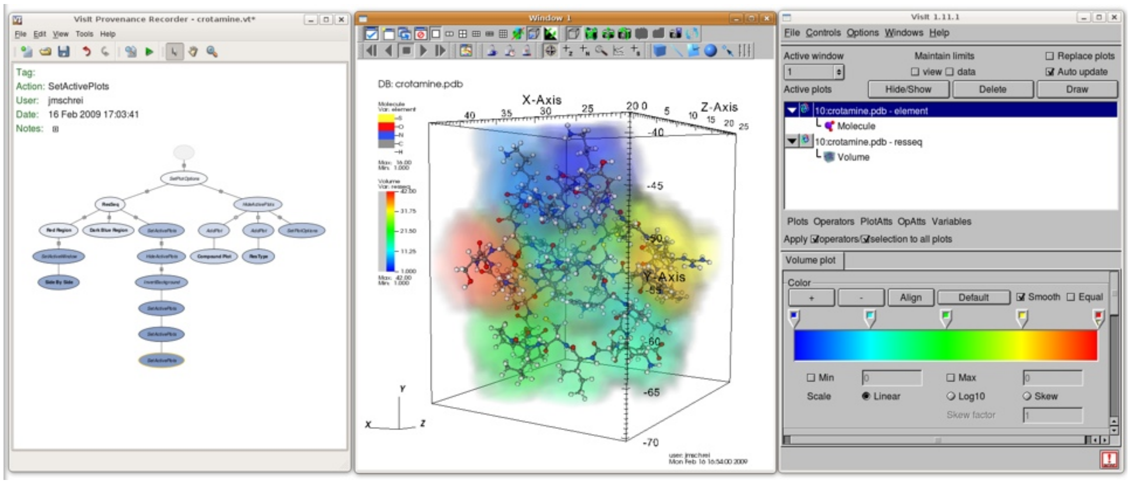
4.8 Reeb Graph Support in VTK

This project aims at enriching the Open Source project Visualization Tool Kit (VTK) with new data analytics capabilities centered on topological methods based on Reeb graphs.

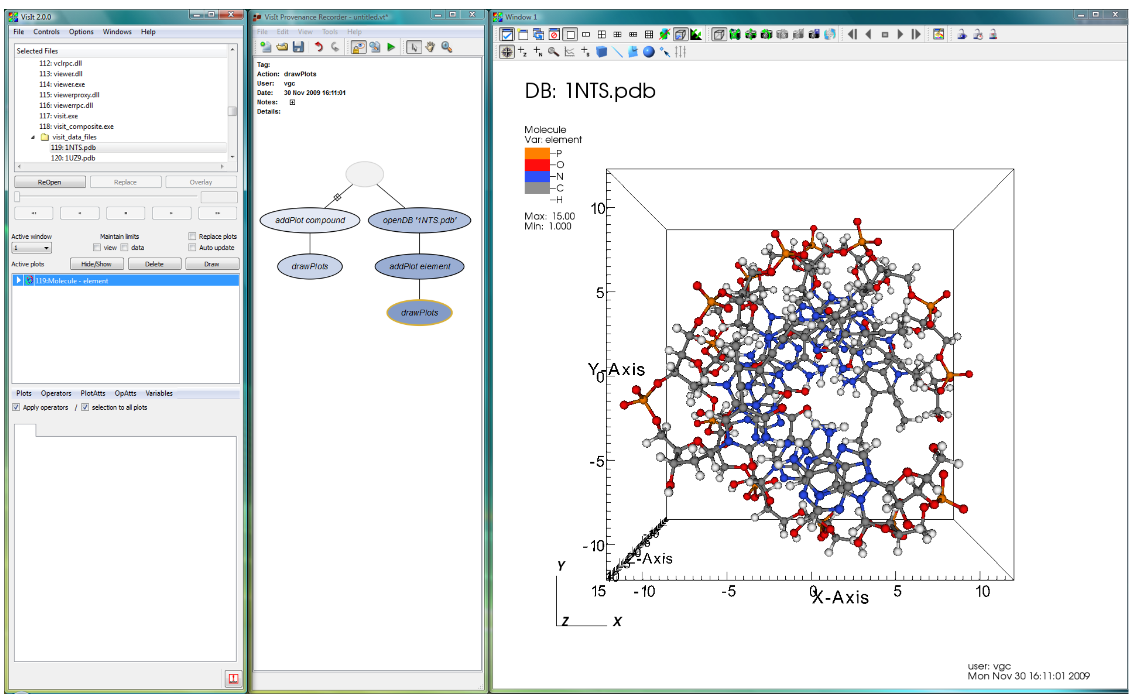
The key element of this project is the implementation of a Reeb graph computation algorithm into VTK (vtkReebGraph class), as well as Reeb-graph based analytics capabilities (for surfaces and volumes) accessible through VTK filters.

Project milestones:

- September 2009. Implementation of the core class vtkReebGraph. Reeb graph computation for manifold surface meshes.
- October 2009. Enrichment of the vtkReebGraph class with low-level streaming capabilities. Streaming Reeb graph computation for manifold surface meshes.
- November 2009. Reeb graph computation for non-manifold surface meshes. Streaming Reeb graph computation for manifold surface meshes.
- December 2009. Reeb graph computation for tetrahedral meshes.
- January 2010. Representation of the Reeb graph as a VTK graph.
- February 2010. Simplification of the Reeb graph through a VTK filter. Custom simplification metric specification.
- March 2010. Level set signature extraction (VTK filter).
- April 2010. Area/hyper-area metric implementation (surfaces). Volume/hyper-volume metric implementation (volumes).
- May 2010. Skeleton 3D embedding for surfaces. Skeleton 3D embedding for tetrahedral meshes.
- June 2010. Contour tree and merge tree computation.



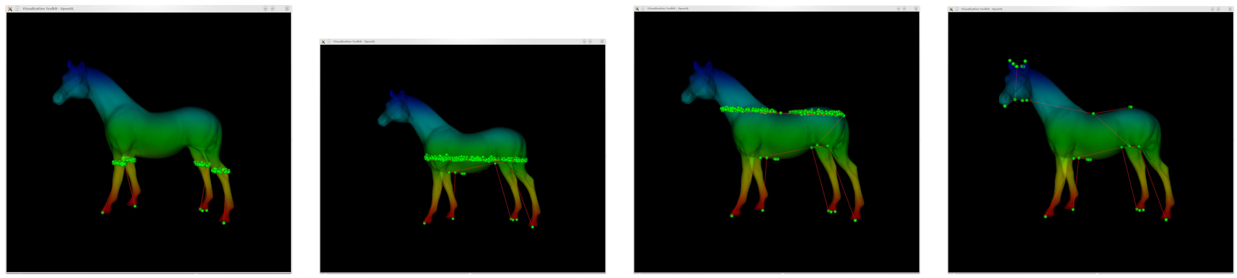
(a) A screenshot of VisTrails plugin for VisIt working on Ubuntu/Linux.



(b) A screenshot of VisTrails plugin for VisIt working on Windows/Vista.

Figure 24: Screenshots showing the VisTrails plugin for VisIt working Linux (top) and Windows (bottom).

.



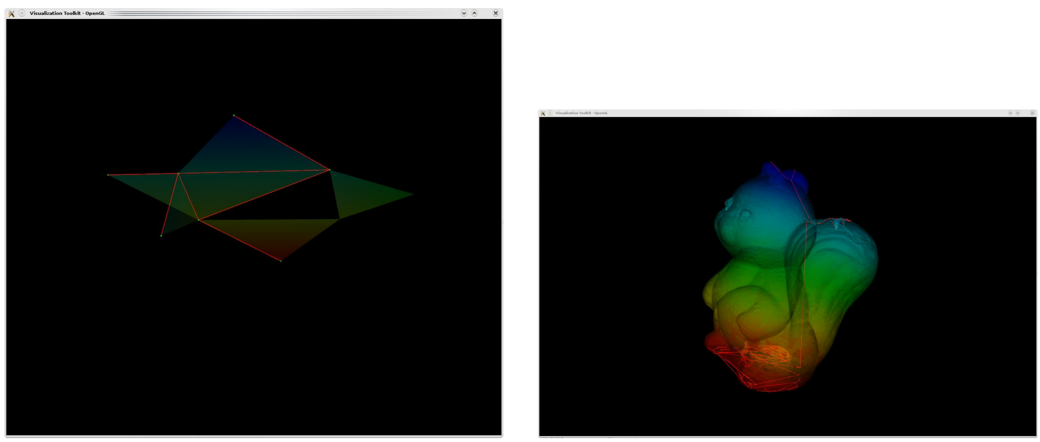
(a) Processing after 25% of the stream.

(b) After 50% of the stream.

(c) After 75% of the stream.

(d) After processing all of the stream.

Figure 25: Reeb graph of the height function on the horse surface computed by the new vtkReebGraph class in VTK.



(a) Reeb graph of the height function on a simple, non-manifold surface.

(b) Reeb graph of the height function on a complex, non-manifold surface.

Figure 26: The new vtkReebGraph class can compute the Reeb Graph on non-manifold surfaces, both simple (left) and complex (right).

- July 2010. Contour tree and merge tree simplification.
- August 2010. Contour tree computation from K-neighbors on point sets.
- September 2010. Study on the integration of time-dependent data.

The first three milestones have been reached during this period. They are computing the Reeb graph for manifold surface meshes (Figure 4.8(d)), enable computation of the Reeb graph on manifold surfaces in a streaming processing model (Figure 4.8(a)–(d)), Reeb graph computations on non-manifold surfaces (Figure 4.8).

5 Publications

5.1 Peer-reviewed Journal Articles

1. John C. Anderson, Luke Gosink, Mark A. Duchaineau, and Ken Joy. Interactive visualization of function fields by range-space segmentation. *Computer Graphics Forum*, 28(3):727–734, June 2009.

2. Brian C. Budge, Tony Bernardin, Shubhabrata Sengupta, Ken Joy, and John D. Owens. Out-of-core data management for path tracing on hybrid resources. In *Proc. Eurographics 2009 (to appear)*, volume 28, pages 385–396, March 2009.

3. Hari Krishnan, Christoph Garth, and Ken Joy. Time and streak surfaces for flow visualization in large time-varying data sets. *Proceedings of IEEE Visualization '09*, pages 1267–1274, October 2009.

4. George Ostrouchov. A matrix computation view of fastmap and robustmap dimension reduction algorithms. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1351–1360, 2009.

5. T. Etiene, C. Scheidegger, L. G. Nonato, R. M. Kirby, and C. Silva. Verifiable visualization for isosurface extraction. *IEEE Transaction on Visualization and Computer Graphics*, 16(6), 2009.

6. E. Santos, L. Lins, J. Ahrens, J. Freire, and C. Silva. Vismashup: Streamlining the creation of custom visualization applications. *IEEE Transaction on Visualization and Computer Graphics*, 16(6), 2009.

7. Julien Tierny, Attila Gyulassy, Eddie Simon, and Valerio Pascucci. Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees. *IEEE Transaction on Visualization and Computer Graphics*, 16(6):1177–1184, November 2009.

8. A. Gyulassy, L. G. Nonato, P.-T. Bremer, C. Silva, and V. Pascucci. Robust topology-based multiscale analysis of scientific data. *Computing in Science and Engineering*, 11(5):88–95, 2009.

9. M. Day, J. Bell, P.-T. Bremer, V. Pascucci, V. Beckner, and M. Lijewski. Turbulence effects on cellular burning structures in lean premixed hydrogen flames. *Combustion and Flame*, 156:1035–1045, 2009.

10. E. W. Bethel, C. Johnson, S. Ahern, J. Bell, P.-T. Bremer, H. Childs, E. Cormier-Michel, M. Day, E. Deines, T. Fogal, C. Garth, C. G. R. Geddes, H. Hagen, B. Hamann, C. Hansen, J. Jacobsen, K. Joy, J. Krüger, J. Meredith, P. Messmer, G. Ostrouchov, V. Pascucci, K. Potter, Prabhat, D. Pugmire, O. Rübel, A. Sanderson, C. Silva, D. Ushizima, G. Weber, B. Whitlock, and K. Wu. Occam's razor and petascale visual data analysis. *Journal of Physics Conference Series, Proceedings of SciDAC 2009*, 180, June 2009. LBNL-2210E.

11. Peer-Timo Bremer, Gunther H. Weber, Valerio Pascucci, Marcus S. Day, and John B. Bell. Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics*, 2009. LBNL-2276E. doi:10.1109/TVCG.2009.69. In press.

12. Scott E. Dillard, Vijay Natarajan, Gunther H. Weber, Valerio Pascucci, and Bernd Hamann. Topology-guided tessellation of quadratic elements. *International Journal of Computational Geometry & Applications (IJCGA)*, 19(2):195–211, April 2009. A preliminary version of this paper appeared in Proceedings of the 17th International Symposium on Algorithms and Computation, Kolkata, India, Lecture Notes in Computer Science (LNCS) 4288, 2006, 722–731. LBNL-63771.

13. Oliver Rübel, Cameron G. R. Geddes, Estelle Cormier-Michel, Kesheng Wu, Prabhat, Gunther H. Weber, Daniela M. Ushizima, Peter Messmer, Hans Hagen, Bernd Hamann, and Wes Bethel. Automatic beam path analysis of laser wakefield particle acceleration data. *IOP Computational Science & Discovery*, 2(015005 (38pp)), November 2009. LBNL-2734E.

14. Gunther H. Weber, Oliver Rübel, Min-Yu Huang, Anagla H. DePace, Charless C. Fowlkes, Soile V.E. Keränen, Cristian L. Luengo Hendriks, Hans Hagen, David W. Knowles, Jitendra Malik, Mark D. Biggin, and Bernd Hamann. Visual exploration of three-dimensional gene expression using physical views and linked abstract views. *IEEE Transactions on Computational Biology and Bioinformatics*, 6(2):296–309, April-June 2009. doi: 10.1109/TCBB.2007.70249. LBNL-63776.

15. K. Wu, S. Ahern, E. W. Bethel, J. Chen, H. Childs, E. Cormier-Michel, C. G. R. Geddes, J. Gu, H. Hagen, B. Hamann, W. Koegler, J. Laurent, J. Meredith, P. Messmer, E. Otoo, V. Perevoztchikov, A. Poskanzer, Prabhat, O. Rübel, A. Shoshani, A. Sim, K. Stockinger, G. Weber, and W.-M. Zhang. Fastbit: Interactively searching massive data. *Journal of Physics Conference Series, Proceedings of SciDAC 2009*, 180, June 2009. LBNL-2164E.

16. Oliver Rübel, Gunther H. Weber, Min-Yu Huang, E. Wes Bethel, Mark D. Biggin, Charless. C. Fowlkes, C. Luengo Hendriks, Soile. V. E. Keränen, Michael B. Eisen, David W. Knowles, Jitendra Malik, Hans Hagen, and Bernd Hamann. Integrating data clustering and visualization for the analysis of 3d gene expression data. *IEEE Transactions on Computational Biology and Bioinformatics*, 2010. LBNL-382E (published online, to appear in print).

17. A. Knoll, I. Wald, and C.D. Hansen. Coherent multiresolution isosurface ray tracing. *The Visual Computer*, 25(3):209–225, 2009.

18. A. Knoll, Y. Hijazi, C.D. Hansen, I. Wald, and H. Hagen. Fast ray tracing of arbitrary implicit surfaces with interval and affine arithmetic. *Computer Graphics Forum*, 28(1):26–40, 2009.

19. Aaron Knoll, Younis Hijazi, Rolf Westerteiger, Mathias Schott, Charles Hansen, and Hans Hagen. Volume ray casting with peak finding and differential sampling. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1571–1578, 2009.

20. C. Deitrich, C.E. Scheidegger, J. Schreiner, J. Comba, L.P. Nedel, and C.T. Silva. Edge transformations for improving mesh quality of marching cubes. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):150–159, Sept/Oct 2009.

21. J. Brouillat, C. Bouville, B. Loos, C.D. Hansen, and K. Bouatouch. A bayesian monte carlo approach to global illumination. *Computer Graphics Forum Journal, Early View*, October 2009.

22. Y. Wan, H. Otsuna, C.-B. Chien, and C.D. Hansen. An interactive visualization tool for multi-channel confocal microscopy data in neurobiology research. *IEEE Transactions on Visualization and Computer Graphics, Proceedings of the 2009 IEEE Visualization Conference*, 15(6):1489–1496, Sept/Oct 2009.

5.2 Peer-reviewed Conference Proceedings

1. Emanuele Santos, Julien Tierny, Ayla Khan, Brad Grimm, Lauro Lins, Juliana Freire, , Valerio Pascucci, Claudio Silva, Scott A. Klasky, Roselyne D. Barreto, and Norbert Podhorszki.

Enabling advanced visualization tools in a simulation monitoring system. In *Proceedings of the 5th IEEE International Conference on e-Science*, 2009.

2. Luke Gosink, Kesheng Wu, E. Wes Bethel, John D. Owens, and Kenneth I. Joy. Data parallel bin-based indexing for answering queries on multi-core architectures. In *21st International Conference on Scientific and Statistical Database Management (SSDBM)*, volume 5566, pages 110–129, June 2009. LBNL-2211E.

3. E. Santos, D. Koop, H.T. Vo, E. Anderson, J. Freire, and C.T. Silva. Using workflow medleys to streamline exploratory tasks. In *21st International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 292–301, 2009.

4. T. Ellkvist, D. Koop, J. Freire, C.T. Silva, and L. Stromback. Using mediation to achieve provenance interoperability. In *Proceedings of the IEEE International Workshop on Scientific Workflows, 2009*, page (to appear), 2009.

5. T. Etiene, C.E. Scheidegger, L.G. Nonato, R. Kirby, and C.T. Silva. Verifiable visualization for isosurface extraction. In *IEEE Transactions on Visualization and Computer Graphics, Proceedings of the 2009 IEEE Visualization Conference*, page (accepted), Sept/Oct 2009.

6. E. Santos, L. Lins, J. Ahrens, J. Freire, and C.T. Silva. User-driven application development. In *IEEE Transactions on Visualization and Computer Graphics, Proceedings of the 2009 IEEE Visualization Conference*, page (accepted), Sept/Oct 2009.

7. Peer-Timo Bremer, Gunther H. Weber, Julien Tierny, Valerio Pascucci, Marcus S. Day, and John B. Bell. Exploring large scale turbulent combustion. In *Proceedings of the 5th IEEE International Conference on e-Science*, Oxford, UK, December 2009.

8. Dave Pugmire, Hank Childs, Christoph Garth, Sean Ahern, and Gunther H. Weber. Scalable computation of streamlines on very large datasets. In *Proc. Supercomputing SC09*, Portland, OR, USA, November 2009.

9. S. Kamil, C. Chan, S. Williams, L. Oliker, J. Shalf, M. Howison, E. W. Bethel, and Prabhat. A Generalized Framework for Auto-tuning Stencil Computations. In *Proceedings of Cray User Group Conference*, Atlanta, GA, USA, May 2009. LBNL-2078E, Best Paper Award Winner.

10. Gunther H. Weber, Sean Ahern, E. Wes Bethel, Sergey Borovikov, Hank R. Childs, Eduard Deines, Christoph Garth, Hans Hagen, Bernd Hamann, Kenneth I. Joy, Daniel Martin, Jeremy Meredith, Prabhat, Dave Pugmire, Oliver Rübel, Brian Van Straalen, and Kesheng Wu. Recent advances in visit: Amr streamlines and query-driven visualization. In *Numerical Modeling of Space Plasma Flows: Astronum-2009 (Astronomical Society of the Pacific Conference Series)*, 2010. To appear.

11. Eduard Deines, Martin Hering-Bertram, Jan Mohring, Jevgenijs Jegorovs, and Hans Hagen. Audio-visual virtual reality system for room acoustics. In *Dagstuhl Seminar Scientific Visualization 05-07*, Wadern, Germany, 2009.

12. Kristin Potter, Andrew Wilson, Peer-Timo Bremer, Dean Williams, Charles Doutriaux, Valerio Pascucci, , and Chris R. Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *IEEE Workshop on Knowledge Discovery from Climate Data: Prediction, Extremes, and Impacts*, 2009.

13. M. Schott, V. Pegoraro, C.D. Hansen, K. Boulanger, and K. Bouatouch. A directional occlusion shading model for interactive direct volume rendering. *Computer Graphics Forum (Proceedings of Eurographics/IEEE VGTC Symposium on Visualization 2009)*, 28(3):855–862, 2009.

5.3 Invited Articles

1. Christoph Garth, Eduard Deines, Kenneth I. Joy, E. Wes Bethel, Hank Childs, Gunther Weber, Sean Ahern, Dave Pugmire, Allen Sanderson, and Chris Johnson. Twists and Turns: Vector Field Visual Data Analysis for Petascale Computational Science. *SciDAC Review*, 15, Winter 2009.

2. C. G. R. Geddes, E Cormier-Michel, E. H. Esarey, C. B. Schroeder, J.-L. Vay, W. P. Leemans, D. L.. Bruhwiler, J. R. Cary, B. Cowan, M. Durant, P. Hamill, P. Messmer, P. Mullowney, C. Nieter, K. Paul, S. Shasharina, S. Veitzer, G. Weber, O. Rübel, D. Ushizima, Prabhat, E. W.Bethel, and K. Wu. Large fields for smaller facility sources. *SciDAC Review*, 13, 2009. LBNL-2299E.

5.4 Posters

1. Soile V. E. Keränen, Angela DePace, Ann Hammonds, Bill Fisher, Oliver Rübel, Gunther H. Weber, Clara Henriquez, Charless Fowlkes, Cris L. Luengo Hendriks, E Wes Bethel, Hans Hagen, Bernd Hamann, Jitendra Malik, Susan E. Celniker, David W. Knowles, Michael B. Eisen, and Mark D. Biggin. On computational analysis of quantitative, 3d spatial expression in drosophila blastoderm, December 2-6 2009. poster.

5.5 Books

1. Valerio Pascucci, Xavier Tricoche, Hans Hagen, and Julien Tierny, editors. *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*. Mathematics + Visualization. Springer, 2010.

2. Oliver Rübel. *Linking Automated Data Analysis and Visualization with Applications in Developmental Biology and High-energy Physics*, volume 28 of *Schriftenreihe Informatik*. Der Dekan (hrsg), Fachbereich Informatik, Technische Universität Kaiserslautern, December 2009.

5.6 Book Chapters

1. C.D. Hansen, C.R. Johnson, V. Pascucci, and C.T. Silva. Visualization for data-intensive science. In Kristin Tolle Dan Fay and Tony Hey, editors, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, pages 153–163. Microsoft Research, 2009.

2. A. Gyulassy, P.-T. Bremer, V. Pascucci, and B. Hamann. *Practical Considerations in Morse-Smale Complex Computation*. Mathematics and Visualization. Springer, 2010. to appear.

3. G. Weber, P.-T. Bremer, J. Bell, M. Day, and V. Pascucci. *Feature Tracking Using Reeb graphs*. Mathematics and Visualization. Springer, 2010. to appear.

4. A. Mascarenhas, R. W. Grout, P.-T. Bremer, E. R. Hawkes, V. Pascucci, and J.H. Chen. *Topological feature extraction for comparison of terascale combustion simulation data*. Mathematics and Visualization. Springer, 2010. to appear.

5. Nameeta Shah, Scott E. Dillard, Gunther H. Weber, and Bernd Hamann. *Volume visualization of multiple alignment of large genomic DNA*, pages 325–342. Mathematics and Visualization. Springer-Verlag, Heidelberg, Germany, July 2009. LBNL-63126.

6. Mario Hlawitschkaand Gunther H. Weber, Alfred Anwander, Owen T. Carmichael, Bernd Hamann, and Gerik Scheuermann. *Interactive Volume Rendering of Diffusion Tensor Data*, pages 161–176. Springer-Verlag, Heidelberg, Germany, April 2009.

7. Eduard Deines, Frank Michel, Martin Hering-Bertram, Jan Mohring, and Hans Hagen. *Simulation and Visualization of Indoor Acoustics Using Phonon Tracing*, pages 147–156. Springer-Verlag, Berlin Heidelberg, Germany, August 2009.

8. C.D. Hansen, C.R. Johnson, V. Pascucci, and C.T. Silva. Visualization for data-intensive science. In Kristin Tolle Dan Fay and Tony Hey, editors, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, pages 153–163. Microsoft Research, 2009.

5.7 Theses

1. John C. Anderson. *Scientific Visualization Techniques for Volume Fraction Data and Function Fields*. PhD thesis, University of California, Davis, June 2009.

2. Luke Gosink. *Query-Driven Visualization Strategies for the Analysis and Visualization of Complex Datasets*. PhD thesis, University of California, Davis, December 2009.

5.8 Technical Reports

1. E. Wes Bethel. High performance, three-dimensional bilateral filtering. Technical Report LBNL-1601E, Lawrence Berkeley National Laboratory, Berkeley, CA, USA, 94720, 2009.

2. H.T. Vo, D.K. Osmari, B. Summa, J.L.D. Comba, V. Pascucci, and C.T. Silva. Parallel dataflow scheme for streaming (un)structured data. SCI Technical Report UUSCI-2009-004, SCI Institute, University of Utah, 2009.

3. H.T. Vo, D.K. Osmari, B. Summa, J.L.D. Comba, V. Pascucci, and C.T. Silva. Parallel dataflow scheme for streaming (un)structured data. SCI Technical Report UUSCI-2009-004, SCI Institute, University of Utah, 2009.

4. H.T. Vo and C.T. Silva. Multi-threaded streaming pipeline for vtk. SCI Technical Report UUSCI-2009-005, SCI Institute, University of Utah, 2009.

5.9 Journal Covers

5.9.1 SciDAC Review – Summer 2009

5.9.2 SciDAC Review – Winter 2009

5.10 Press Releases/Articles

1. L. Vu (ed.). NERSC's Deep Sky Project Provides a Portal into Data Universe. Berkeley Lab Computing Sciences News, `http://www.lbl.gov/cs/Archive/news032709a.html`, March 2009.

Figure 27: Massively parallel VORPAL simulations of LOASIS (LBNL) experiments show the structure of a plasma density wave, or wake (blue), driven by an intense laser pulse traveling upward through the image. Simulation by C.G.R. Geddes (LBNL). VisIt visualization by G.H. Weber and C.G.R. Geddes (LBNL). This image appeared on the cover of the Summer 2009 issue of SciDAC Review.

2. L. Vu (ed.). VisIt Now Available on Franklin. Berkeley Lab Computing Sciences News, `http://www.lbl.gov/cs/CSnews/CSnews052609b.html`, May 2009.

3. L. Vu (ed.). NERSC and CRD Help Decipher Science from Compact Accelerator Simulations. Berkeley Lab Computing Sciences News, `http://www.lbl.gov/cs/Archive/news052609a.html`, May 2009.

Figure 28: New visualization techniques developed in VACET provide insight into the complex structures of vector fields. Here, an Integral Surface illustrates vortex formation in the flow field above a wing.

4. R. Black (ed.). Scientists Open Their Eyes to Visualization's Potential. ASCR Discovery Website, `http://ascr-discovery.science.doe.gov/feature/viz1.shtml`, June 2009.

5. L. Vu (ed.). Visualizing the Future of Scientific Discovery. Berkeley Lab Computing Sciences News, `http://www.lbl.gov/cs/Archive/news061109.html`, June 2009.

6. J. Bashor (ed.). DOE Researchers Test Limits of Visualization Tool. HPCwire, `http://www.hpcwire.com/topic/visualization/DOE-Researchers-Test-Limits-of-Visualization-Tool-4753367`

html?page=1, June 2009.

6 Outreach

6.1 Invited Presentations

1. E. Wes Bethel. Foghorns, Lighthouses and the Circuitous, Hazard-laden Path Towards Extreme Scale Data Analysis. In *ICAP '09 – 10th International Computational Accelerator Physics Conference*, San Francisco, CA, September 2009.
2. E. Wes Bethel. The Art and Science of Scientific Visualization. In *LBNL Computer Sciences Summer Student Seminar Series*, Berkeley, CA, July 2009.
3. E. Wes Bethel. Occam's Razor and Petascale Visual Data Analysis. In *SciDAC 2009 Program Conference*, San Diego, CA, June 2009. Featured presentation.
4. Hank Childs. Why petascale visualization will change the rules (keynote presentation). In *REVISE Workshop*, Atlantic City, NJ, October 2009.
5. Hank Childs. Visit, what analysis should we be doing?, and how will we do it with big data? In *Turbulent Mixing and Beyond 2009*, Trieste, Italy, July 2009.
6. Hank Childs. Overview of the visit project. In *CSCADS Workshop 2009*, Lake Tahoe, NV, August 2009.
7. Eduard Deines. Integral curves in block-structured amr simulations. In *Dagstuhl Scientific Visualization Seminar (Seminar 09251)*, Schloss Dagstuhl (Leibniz-Zentrum für Informatik), Wadern, Germany, June 2009.
8. Gunther H. Weber. The contour spectrum revisited. In *Dagstuhl Scientific Visualization Seminar (Seminar 09251)*, Schloss Dagstuhl (Leibniz-Zentrum für Informatik), Wadern, Germany, June 2009.
9. Gunther H. Weber. Visual exploration of turbulent combustion and laser-wakefield accelerator simulations. In *Paul Scherer Institut*, Villigen, Switzerland, June 2009.
10. Gunther H. Weber. Recent advances in visit: Streamlines and query-driven visualization. In *ASTRONUM-2009—the 4th International Conference on Numerical Modeling of Space Plasma Flows*, Chamonix, France, June/July 2009.
11. George Ostrouchov. Fast simultaneous dimension reduction and clustering: Viewing data from extremes. In *SRC 2009: Spring Research Conference on Statistics in Industry and Technology*, Vancouver, Canada, May 2009.
12. Maciej Haranczyk, Gunther H. Weber, and Maciej Gutowski. Visualization of molecular orbitals and the related electron densities. In *The 238th ACS National Meeting*, Washington, DC, August 2009.
13. Valerio Pascucci. Topological data analysis. In *CSCADS Workshop 2009*, Lake Tahoe, NV, August 2009.
14. Valerio Pascucci. Topological data analysis. In *CSRI workshop on workshop on combinatorial algebraic topology*, Santa Fe, NM, August 2009.

15. Chuck Hansen. Vacet: Visualisation and analytics center for enabling technologies. In *International Supercomputing Conference*, Hamburg, Germany, June 2009.

16. Chuck Hansen. Interactive texture-based flow visualization. In *University of Kaiserslautern*, Kaiserslautern, Germany, June 2009.

17. Chuck Hansen. Multi-field volume visualization. In *IAMCS Spring Symposium*, College Station, Texas, USA, May 2009.

18. Chuck Hansen. Large-scale scientific visualization. In *IAMCS Spring Symposium*, College Station, Texas, USA, May 2009.

6.2 Tutorials and Training

1. Gunther H. Weber, Peer-Timo Bremer, Hamish Carr, and Attila Gyulassy. Scalar topology in visual data analysis. IEEE VisWeek 2009, Atlantic City, NJ, USA, October 2009.

2. Hank Childs and Sean Ahern. Visit - visualization and analysis for very large data sets. Supercomputing 2009, Portland, OR, USA, November 2009.

3. Hank Childs and Sean Ahern. Visualization and analysis using visit. IEEE VisWeek 2009, Atlantic City, NJ, USA, October 2009.

4. Christoph Garth, Filip Sadlo, Jens Krüger, Daniel Weiskopf, and Hank Childs. Visualization of time-varying vector fields. IEEE VisWeek 2009, Atlantic City, NJ, USA, October 2009.

5. Hank Childs. Visualization and analysis using visit. Tutorial at the NERSC User's Group Meeting, Boulder, CO, USA, October 2009.

6. Hank Childs. Visualization and analysis using visit. Tutorial at the ACTS Workshop, Berkeley, CA, USA, August 2009.

7. Hank Childs. Visualization and analysis using visit. Tutorial for Princeton/PPPL users, Princeton, NJ, USA, July 2009.

8. Hank Childs and David Pugmire. Visualization and analysis using visit. Tutorial at SciDAC 2009, San Diego, CA, USA, June 2009.

7 Service

7.1 Technical Reviewer

1. DOE Early Career PI, Fall 2009.

2. IEEE Computer Graphics and Applications.

3. IEEE Transactions on Visualization and Computer Graphics.

4. International Journal of High Performance Computing.

5. Concurrency and Computation: Practice and Experience.

6. PacificVis 2009.

7. Astronum 2009.
8. EuroVis 2009.
9. TopoInVis 2009.
10. IEEE Visualization.
11. IEEE InfoVis.
12. IEEE Visual Analytics Science and Technology..
13. EuroGraphics 2009.
14. EuroGraphics 2010.
15. Visualization in Medicine and Life Sciences (VMLS) 2009.

7.2 Program Committee

1. Co-Chair of the ACM Symposium on Computational Geometry 2010..
2. Supercomputing 2009.
3. Supercomputing 2010.
4. IASTED International Conference on Computer Graphics and Imaging, CGIM 2010.
5. Eurographics/IEEE Symposium on Visualization (EuroVis 2010) , June 9 - 11, Bordeaux, France 2010.
6. Program Committee, Organizing Committee, SciDAC 2009.
7. Organizers of VisNight 2009.
8. ACM I3D 2009.
9. Topology in Visualization 2009.
10. Workshop on Novel Computing for Life Sciences, 2010.
11. Pacific Vis 2010.

7.3 Editorial Board

1. Journal of Computational Science, Editorial Board (2009-).
2. Electronic Transactions in Numerical Analysis, Editorial Board (2003-).
3. Visualization and Mathematics, Book Series Editorial Board, Springer-Verlag (1997–).
4. Journal of Computing and Visualization in Science, Editorial Board, (1996–).

7.4 Other

1. Member, NSF Office of Cyberinfrastructure Task Force on Software Infrastructure (2009-).
2. Member, NSF Office of Cyberinfrastructure Task Force on Grand Challenge Communities (2009-).
3. CRA Computing Community Consortium (CCC) (2009-).
4. Computing Research Association Education Committee (2008-).
5. Working Group Co-Chair, DOE Workshop on Scientific Opportunities in Modeling and Simulation at the Extreme Scale for Biological Sciences, 2009.

8 Awards

1. **Best Paper Award.** S. Kamil, C. Chan, S. Williams, L. Oliker, J. Shalf, M. Howison, E. W. Bethel, and Prabhat. A Generalized Framework for Auto-tuning Stencil Computations. In *Proceedings of Cray User Group Conference*, Atlanta, GA, USA, May 2009. LBNL-2078E, Best Paper Award Winner.
2. **Bandwidth Challenge Finalist.** Supercomputing 2009, Portland OR, November 2009.
3. **People's Choice Award.** SciDAC 2009, Visualization Night, San Diego, CA, June 2009. *Simulation of the Turbulent Flow in an Advanced Recycling Nuclear Reactor.* H. Childs, P. Fischer, A. Obabko, D. Pointer, A. Siegel.
4. **People's Choice Award.** SciDAC 2009, Visualization Night, San Diego, CA, June 2009. *ImageVis3D.* J. Krüger, T. Fogal.